



SZÁMÍTÓGÉP ÁLTAL VEZÉRELT ARCFELISMERŐ RENDSZER KIALAKÍTÁSA ПОБУДОВА СИСТЕМИ ЦИФРОВОГО РОЗПІЗНАВАННЯ ОБЛИЧЧЯ

Készítette: Svec Antal

AM III/6

Témavezető: Simon Lénárd

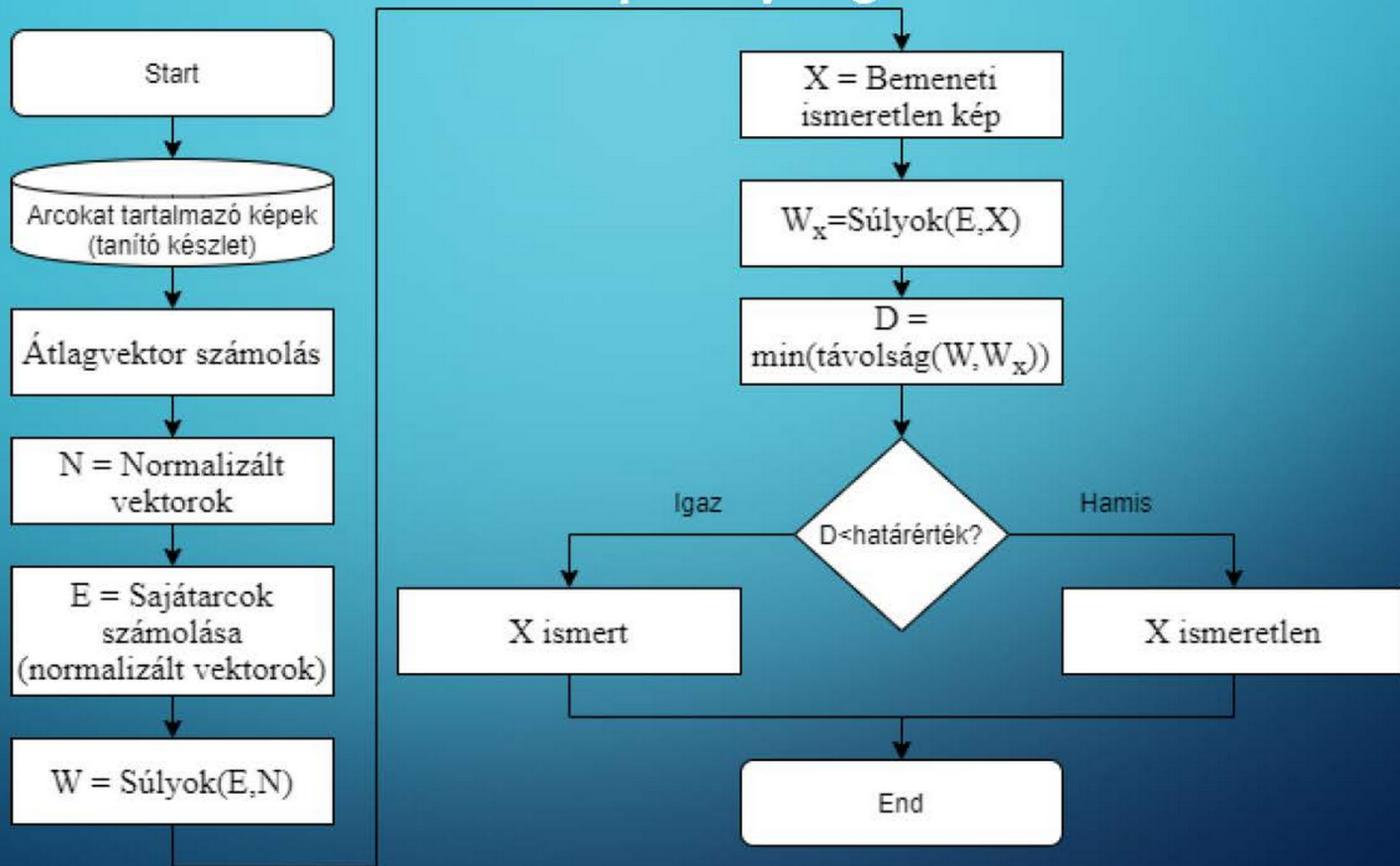
Eigenface algoritmus

Алгоритм власных граней

- Arcfelismerő módszerek típusai:
 - Parametrikus(fotometrikus) – kép vizsgálat, tulajdonságok kinyerése matematikai műveletek segítségével.
 - Eigenface
 - LBPH
 - Fisherfaces
 - Geometriai - az arc különböző részleteinek szem, ajkak, orr, áll stb. - egymáshoz viszonyított elhelyezkedését és méretét elemzik.
- Arcfelismerő rendszer működésének lépései:
 - Adatok gyűjtése
 - Arcdetektálás
 - Jellemzők kinyerése(tanítás)
 - Azonosítás

Eigenface algoritmus folyamat ábra:

Блок-схема алгоритму Eigenface:



Eigenface algoritmus tanítási folyamata:

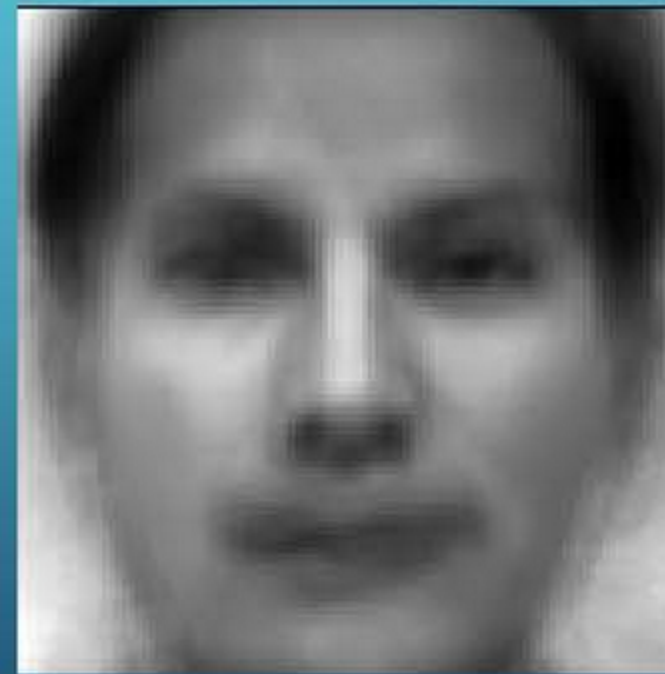
Процес навчання алгоритму Eigenface:

1. Elkészítem a tanító készletet.



2. Kiszámolom az átlagarcot

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$



3. Normalizálom a tanító készletet

$$\phi_i = x_i - \mu$$



4. Kiszámolom a sajátarcokat (kovariáns mátrix, sajátértékek, sajátvektorok)

$$C = \frac{1}{N} \sum_{i=1}^N \phi_i \phi_i^T = AA^T$$

$$f(\lambda) = \det(C - \lambda * E)$$

$$(C - \lambda_i * E) * \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$



Rendszer megalkotása Створення системи

- Program rész

- Rendszer vezérléshez megalkottam a vezérlő felületet.
- A rendszer képes:
 - Új felhasználót beregisztálni
 - Elvégezni a felismerési algoritmus tanítási folyamatot
 - Élőben működtetni az arcfelismerést és azonosítást
 - Törölni a felhasználót a rendszerből
 - Kikapcsolódni

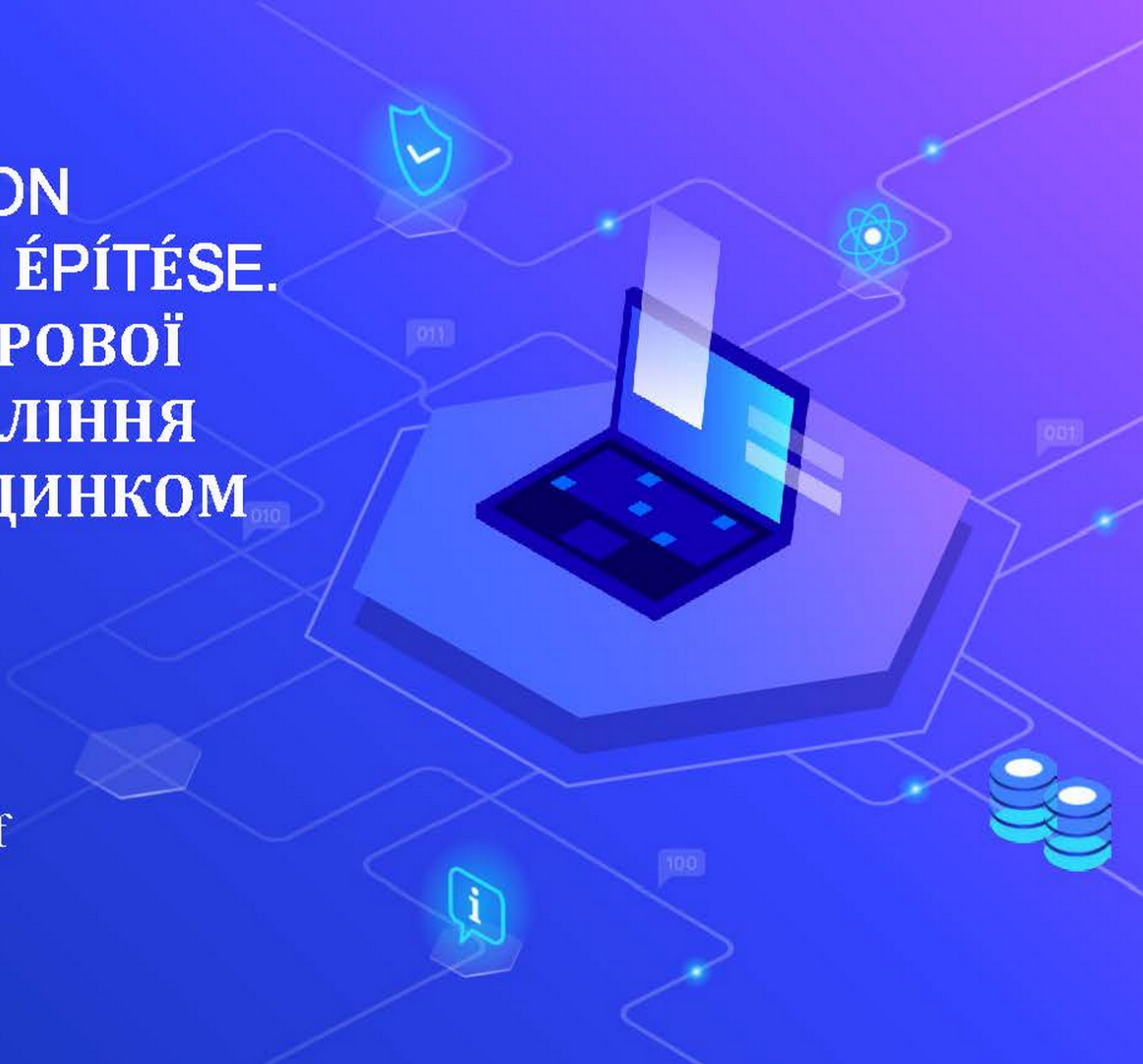
- Eszköz rész

- Rendszer vezérléshez használtam kijelzőt és gombokat
- A képek készítéséért RaspberryPi kamera felelős volt
- Terveztem tokot az eszköz számára a Fusion360 nevezetű alkalmazásba
- 3D nyomtató segítségével átvittem valóságba



OKOS OTTHON RENDSZERÉNEK KI ÉRÍTÉSE. ПОБУДОВА ЦИФРОВОЇ СИСТЕМИ УПРАВЛІННЯ „РОЗУМНИМ” БУДИНКОМ

Készítette: Simon István,
AMI/6
Téma vezető: Szocska József



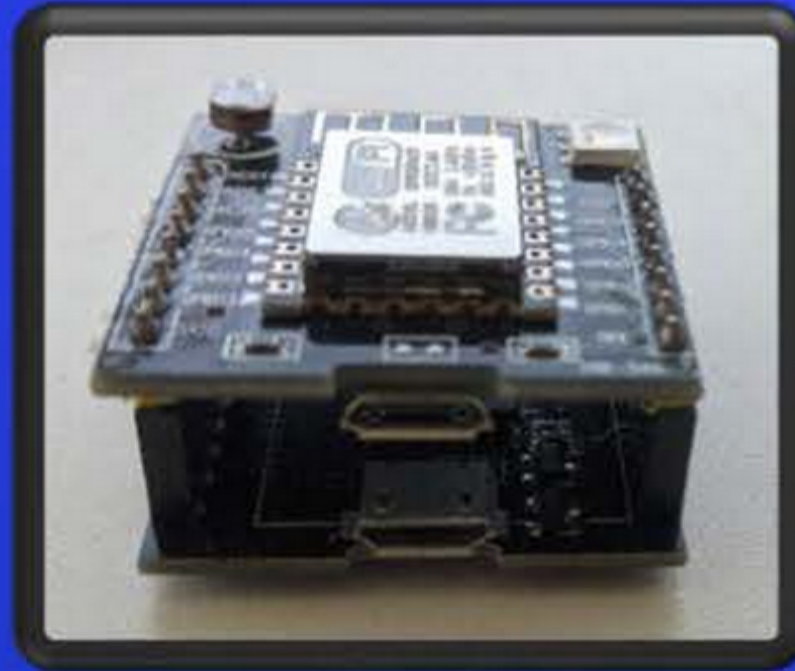
Felhasznált eszközök Використовувані



ESP32 fejlesztőlap



TFT LCD kijelző



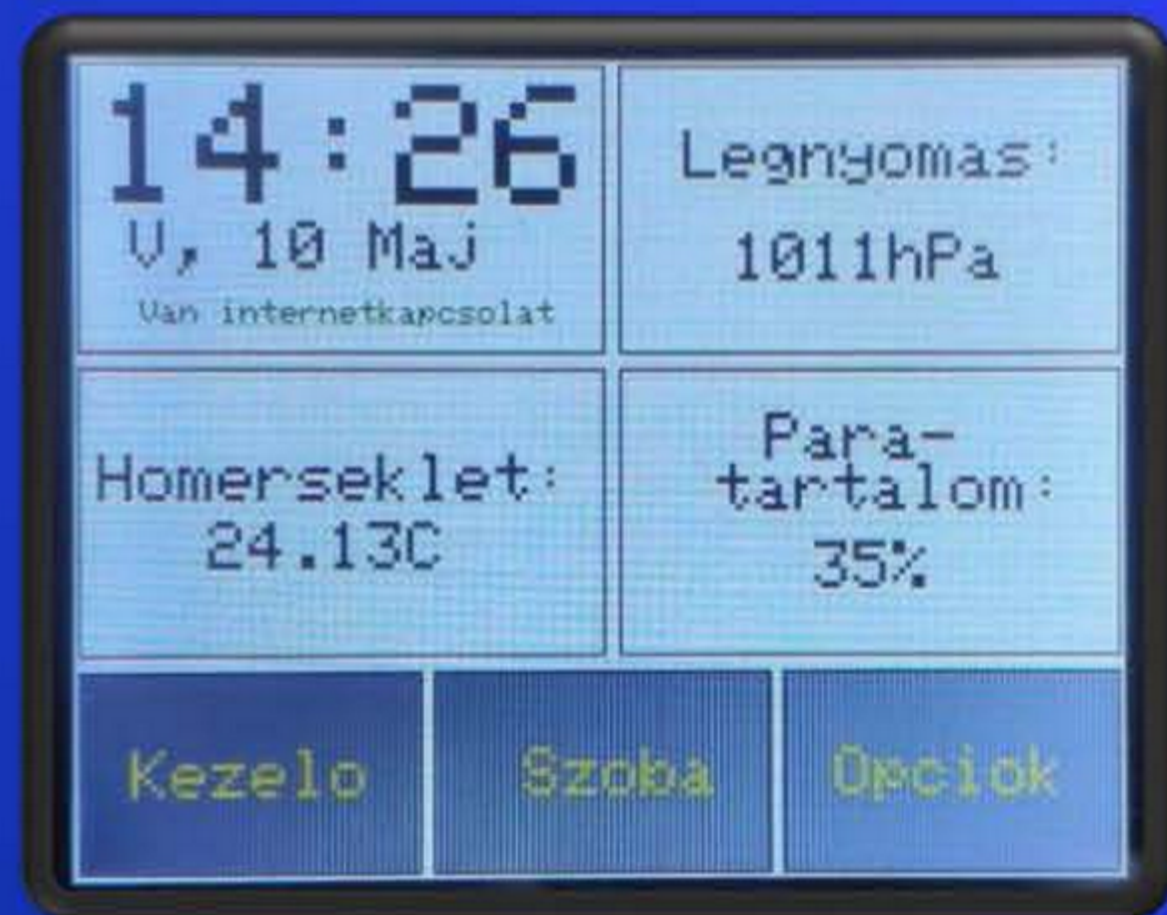
Witty Cloud fejlesztőlap

A vezérlő egység felépítése

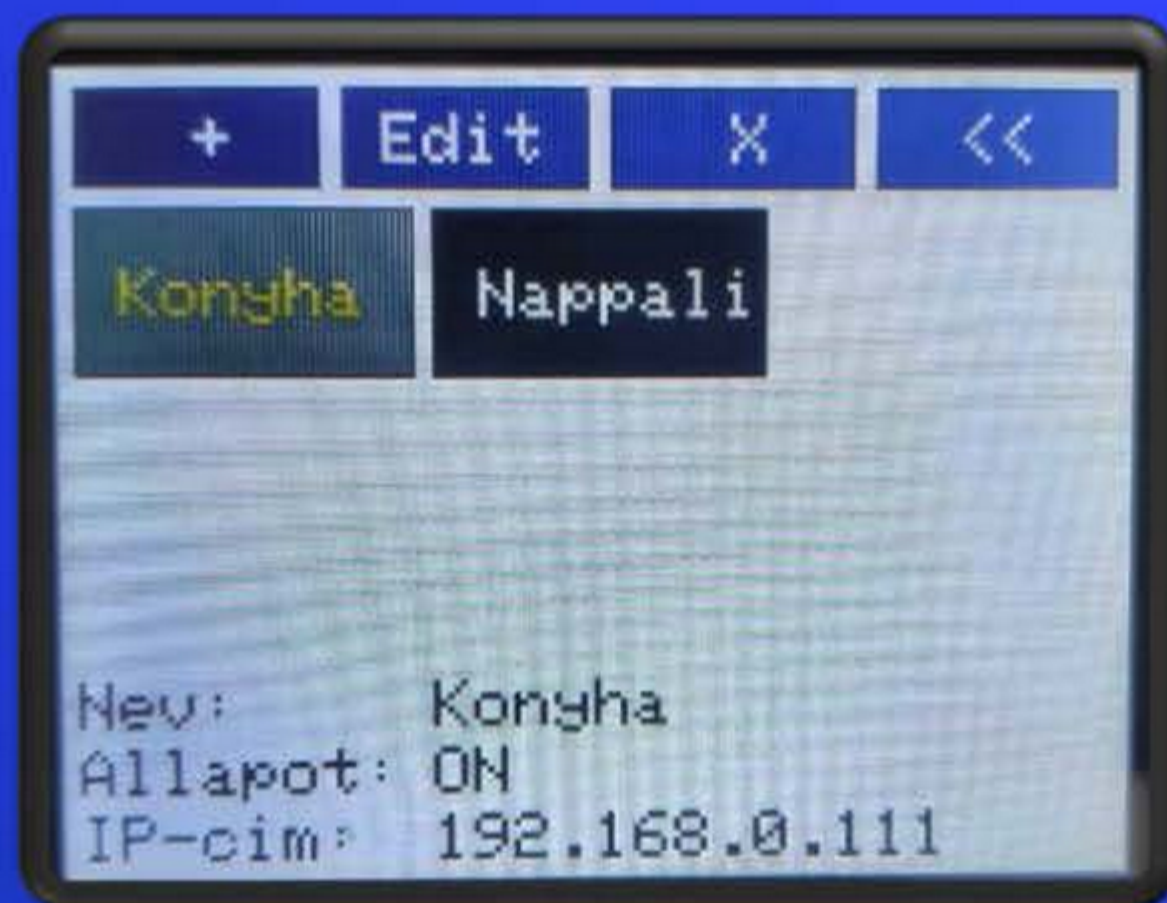
Побудова блока керування



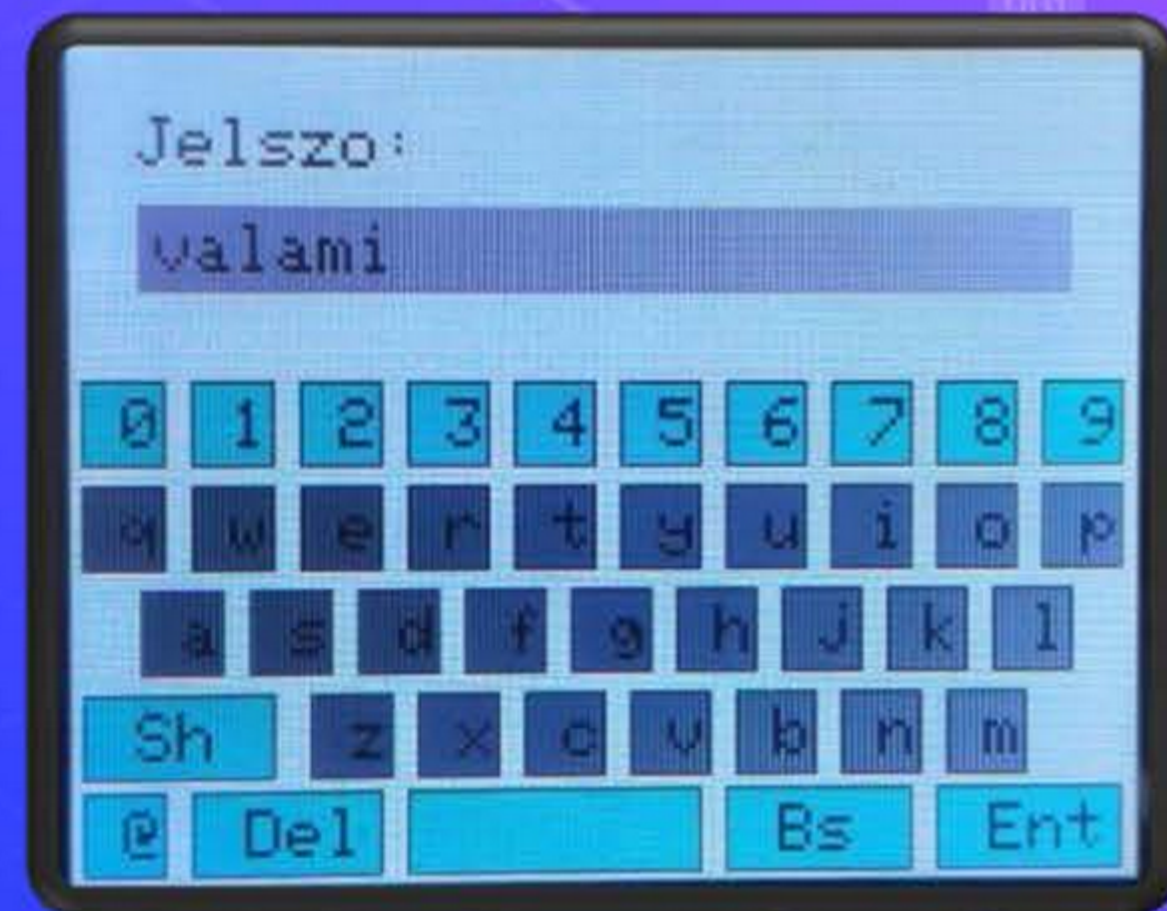
Felületek Поверхні



Főképernyő



Eszközkezelő



Billeentyűzet

Felületek Поверхни

Lámpa adatok szerkesztése

Valassza ki, melyik adatot szeretne szerkeszteni

Nev	IP-cim	Kesz
Regi nev: Konyha	Regi IP: 192.168.0.111	
Uj nev: Konyha	Uj IP: 192.168.0.111	

IP: 192.168.0.110

Konyha Nappali

Lámpa vezérlés

Opciók

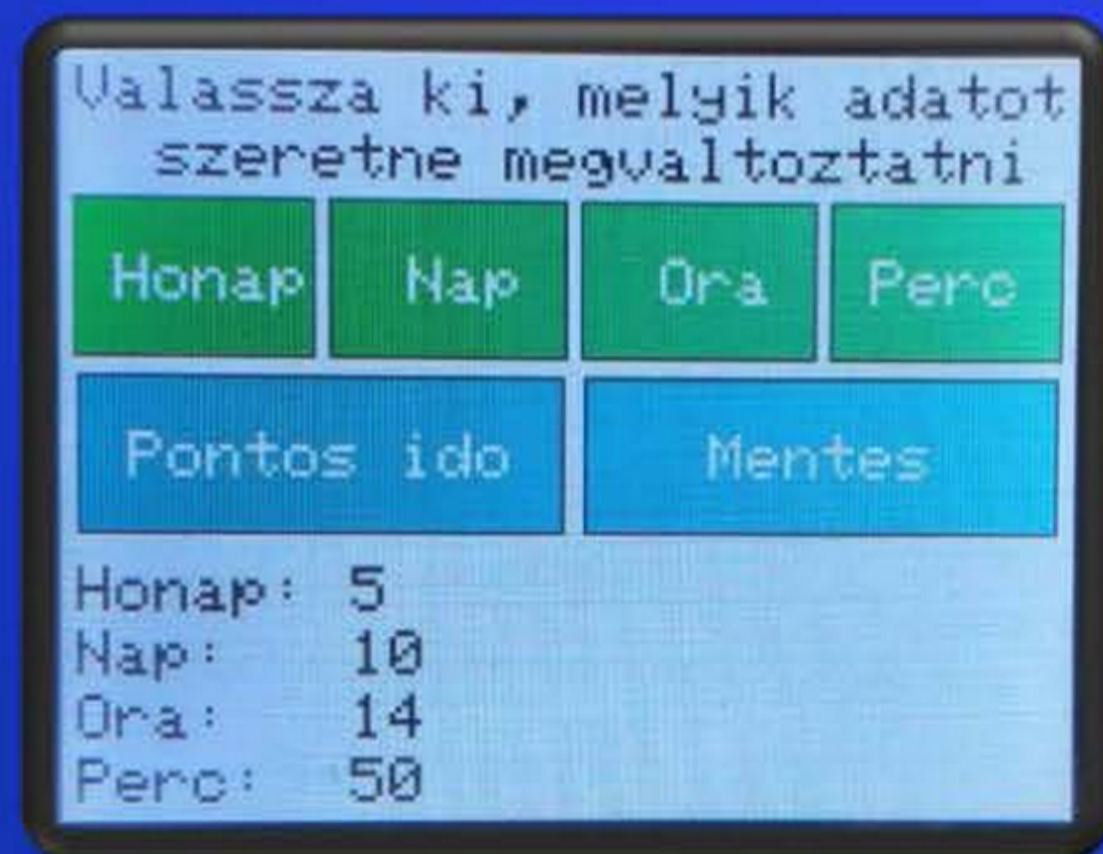
Uissza

Ujrakalibrálás

Datum es Idő

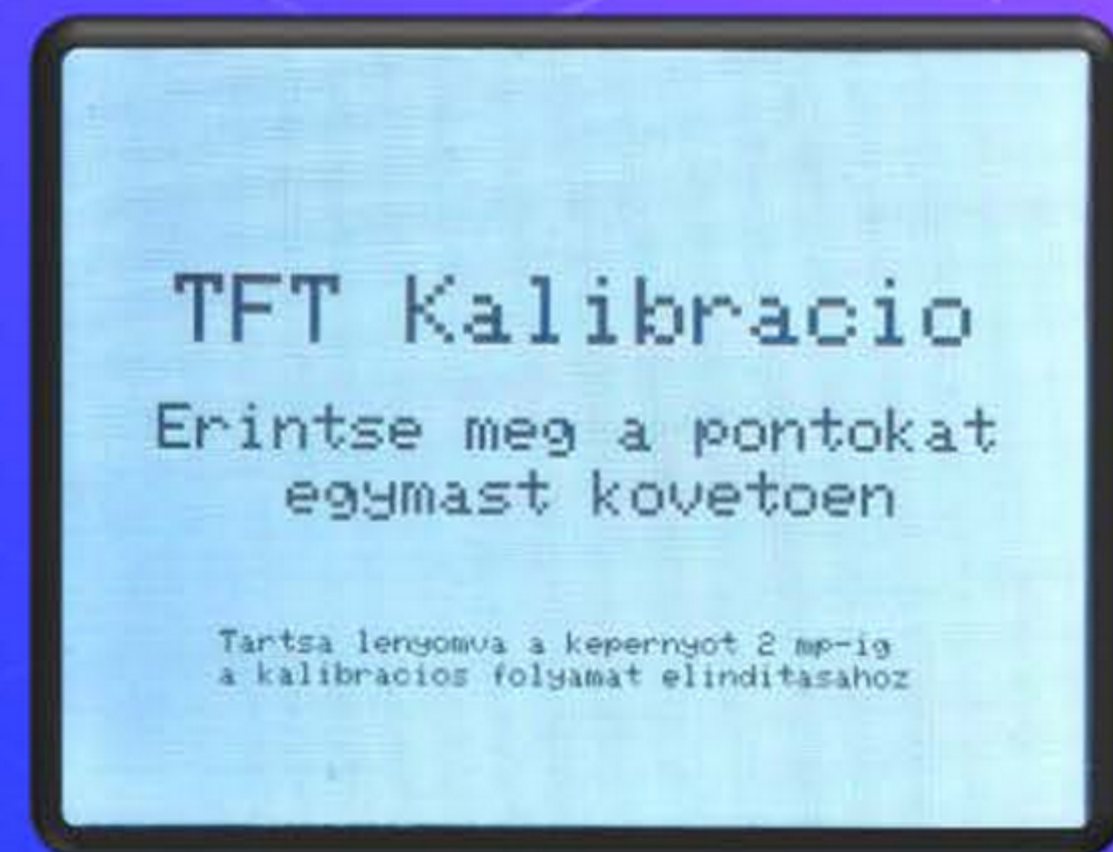
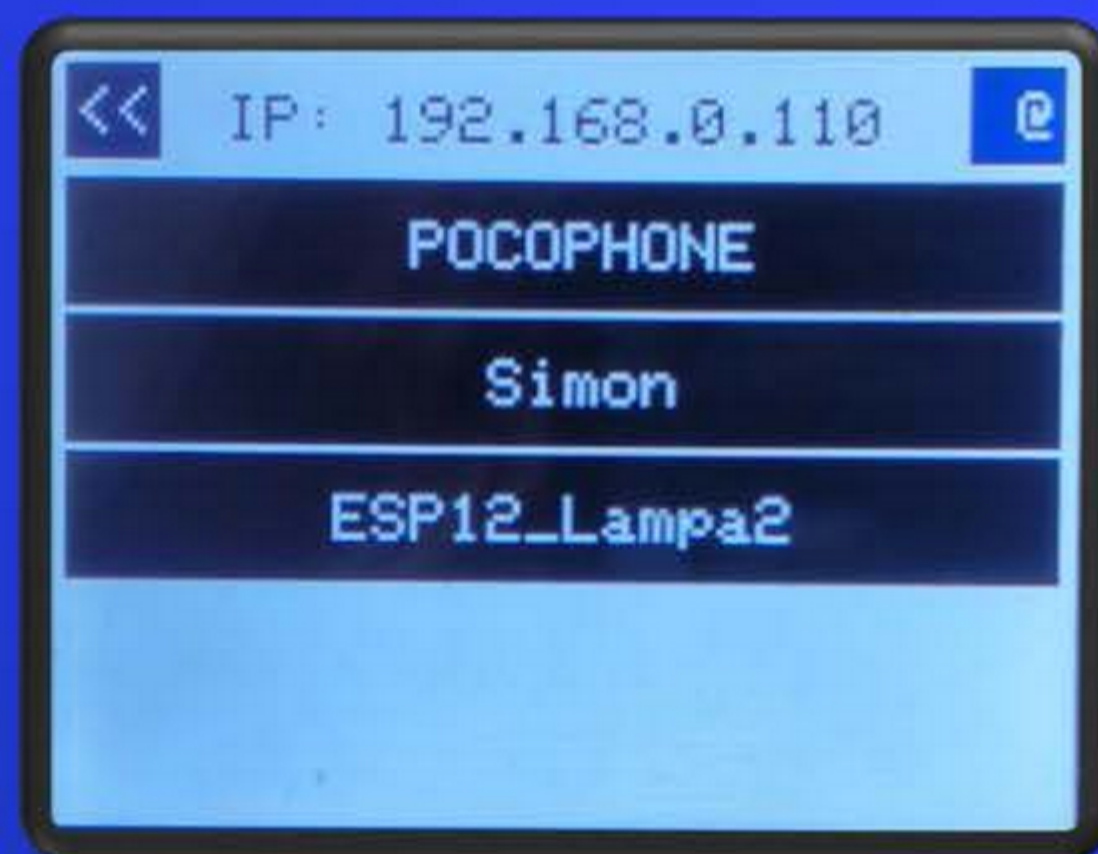
Wifi hálózatok

Felületek Поверхні



Dátum és Idő
beállítása

WiFi hálózatok



Kalibráció elindítása

Felületek Поверхні



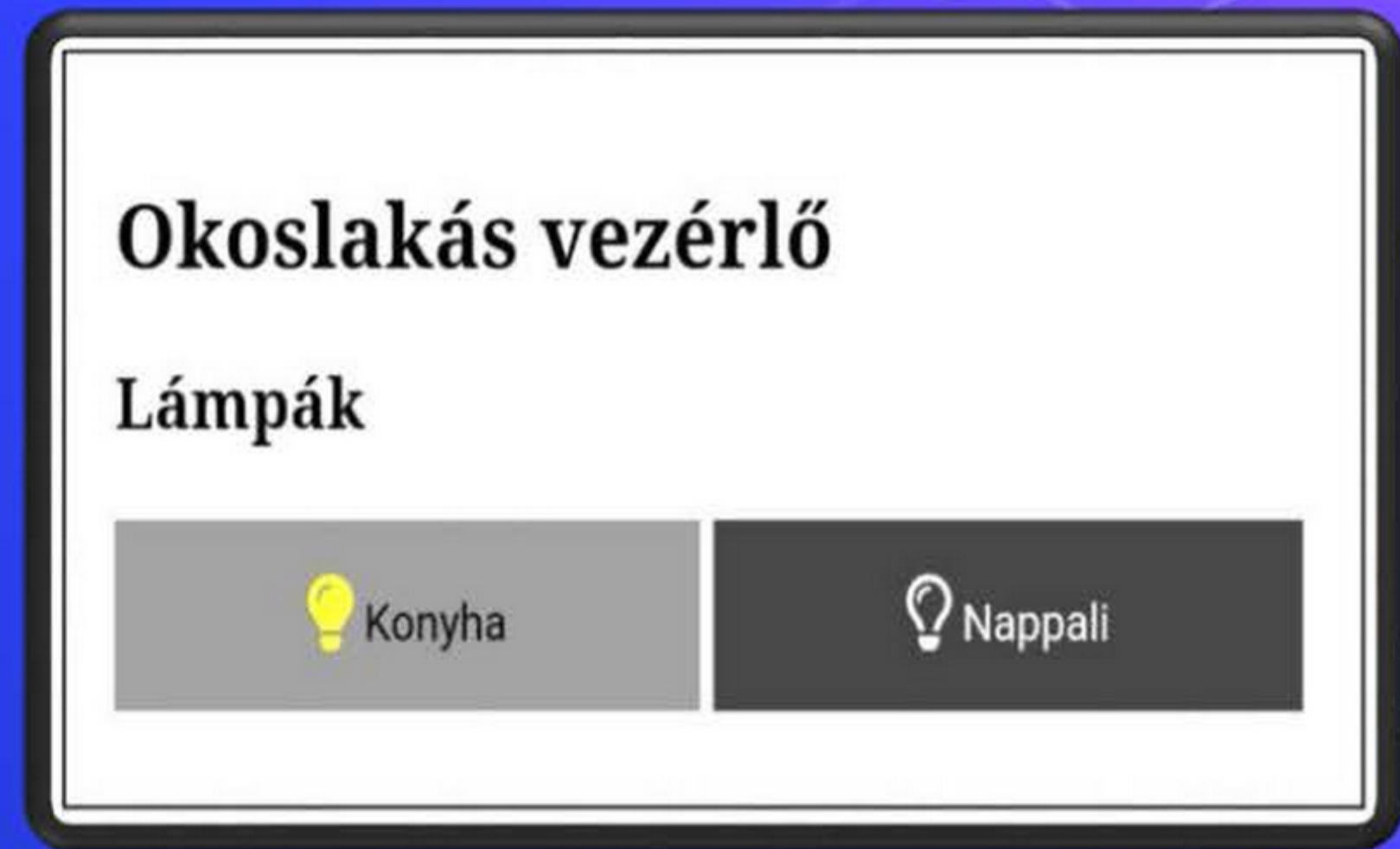
A web browser window showing a form for connecting an ESP12 module to WiFi. The address bar shows '192.168.4.1'. The form has a title 'ESP12 WiFi csatlakozás', two input fields labeled 'SSID' and 'Jelszó', and a 'Submit' button.

ESP12 WiFi csatlakozás

SSID

Jelszó

Eszköz hozzáadása



A web browser window showing a smart home control interface. The title is 'Okoslakás vezérlő'. Below it is the section 'Lámpák' (Lights). There are two buttons: 'Konyha' (Kitchen) with a yellow lightbulb icon, and 'Nappali' (Living Room) with a white lightbulb icon.

Okoslakás vezérlő

Lámpák

Okoslakás vezérlő

**SZEMÉLYES FELHŐRENDSZER KIALAKÍTÁSA
RASPBERRY PI FELHASZNÁLÁSÁVAL.**

**ПОБУДОВА ПЕРСОНАЛЬНОЇ ХМАРНОЇ
СИСТЕМИ ІЗ ЗАСТОСУВАННЯМ RASPBERRY PI**

Készítette: Balla Bianka

AM III/6

Témavezető: Simon Lénárd

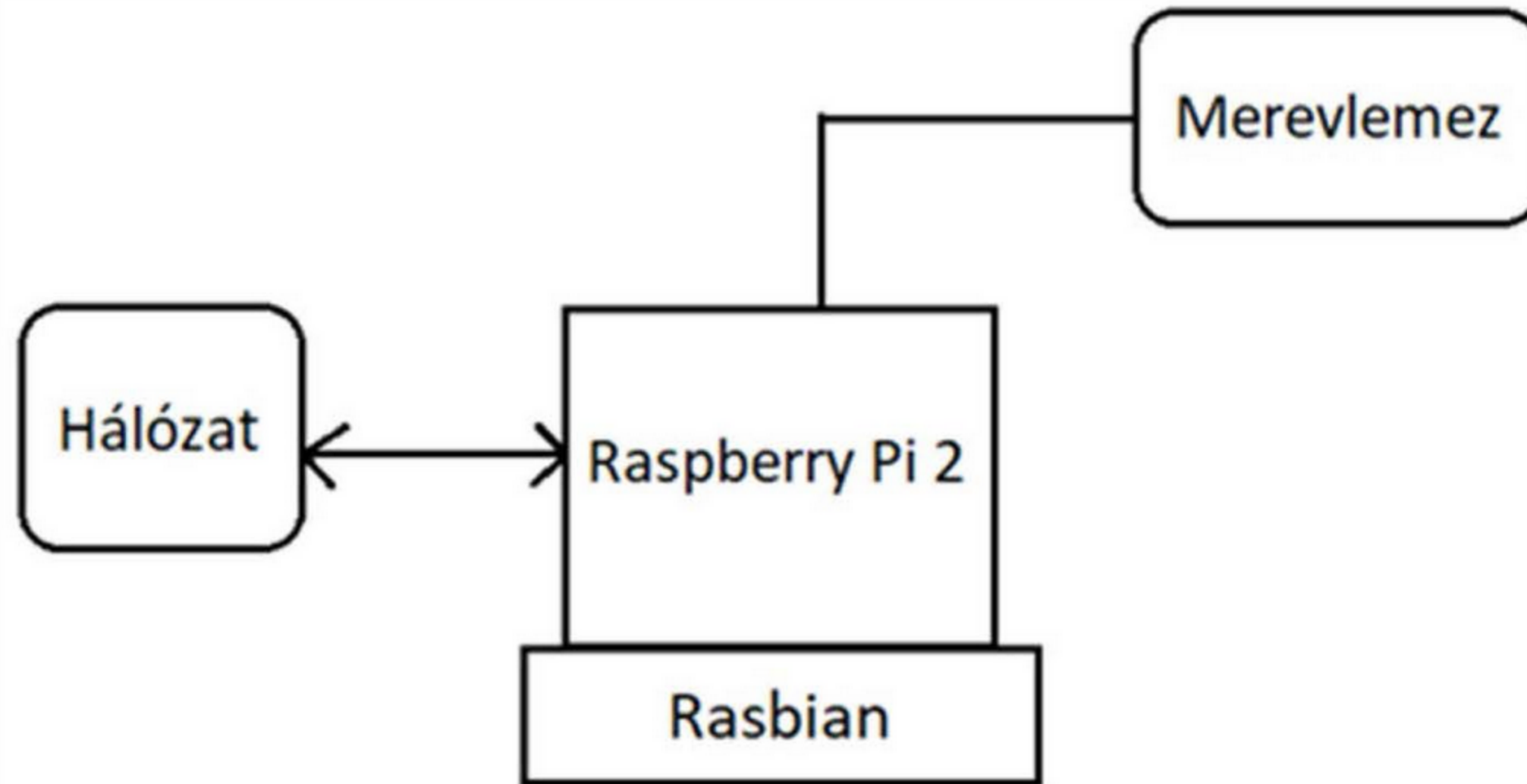
NextCloud

- Nyílt forráskódú szinkronizáló és megosztási szoftver mindenki számára.



A rendszer tervekrajza

План системи

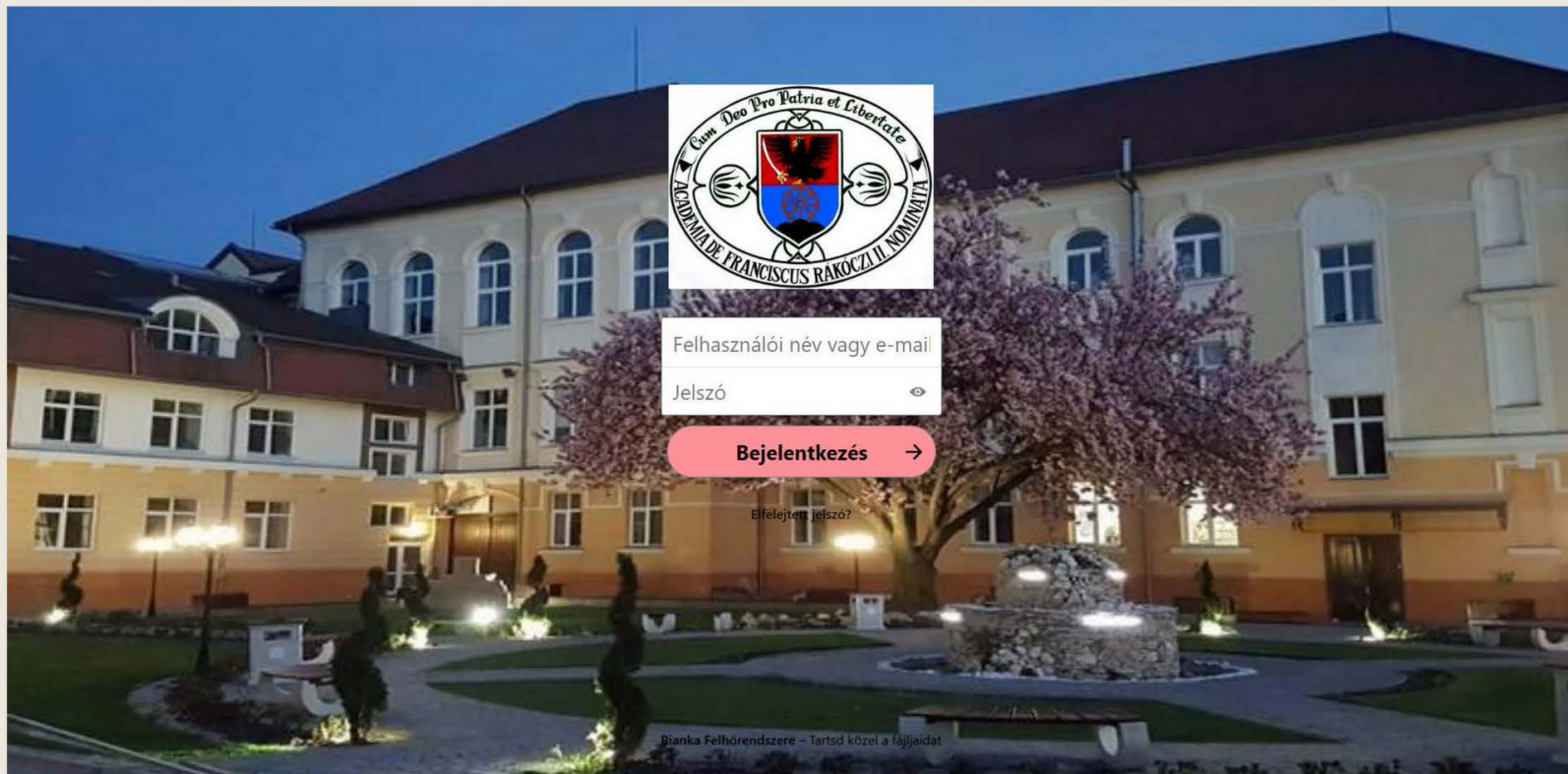


Felhasználók és csoportok

Користувачі та групи

Bejelentkezési felület

Интерфейс входа





GRAFIKUS FELHASZNÁLÓI FELÜLETTEL
RENDELKEZŐ SZOFTVER KÉSZÍTÉSE
MÁTRIXARITMETIKAI MŰVELETEK VÉGZÉSÉRE

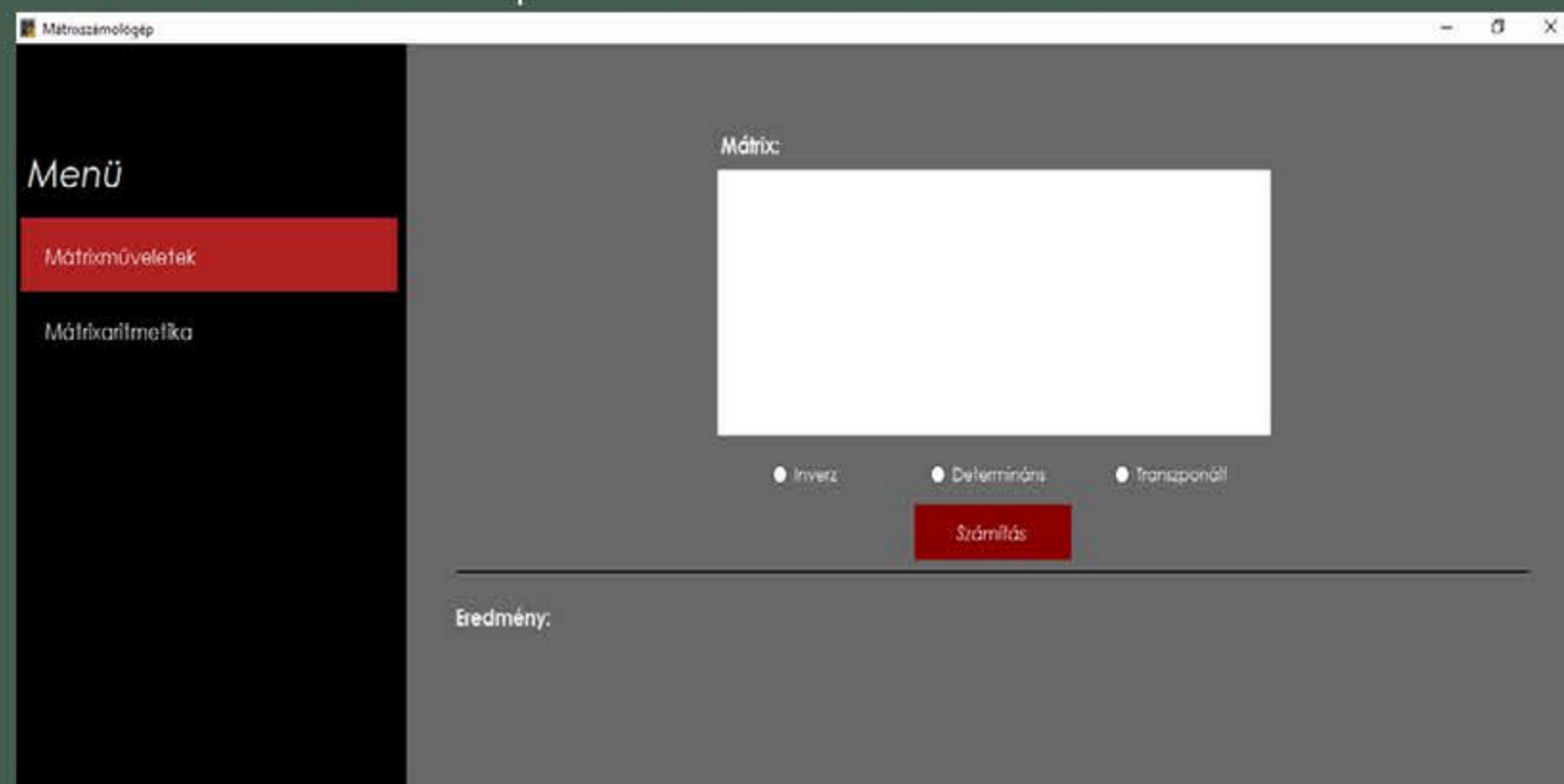
РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ З
ГРАФІЧНИМ ІНТЕРФЕЙСОМ ДЛЯ РЕАЛІЗАЦІЇ
ДІЙ МАТРИЧНОЇ АРИФМЕТИКИ.

Készítette: Nagy Mónika,
AM III/3
Témavezető: Molnár Alexandra

Mátrixműveletek

Матричні операції

- Három művelet tartozik bele:
 - Inverz mátrix
 - Determináns
 - Transzponált
- Művelet lépéseinek levezetése



Mátrixműveletek

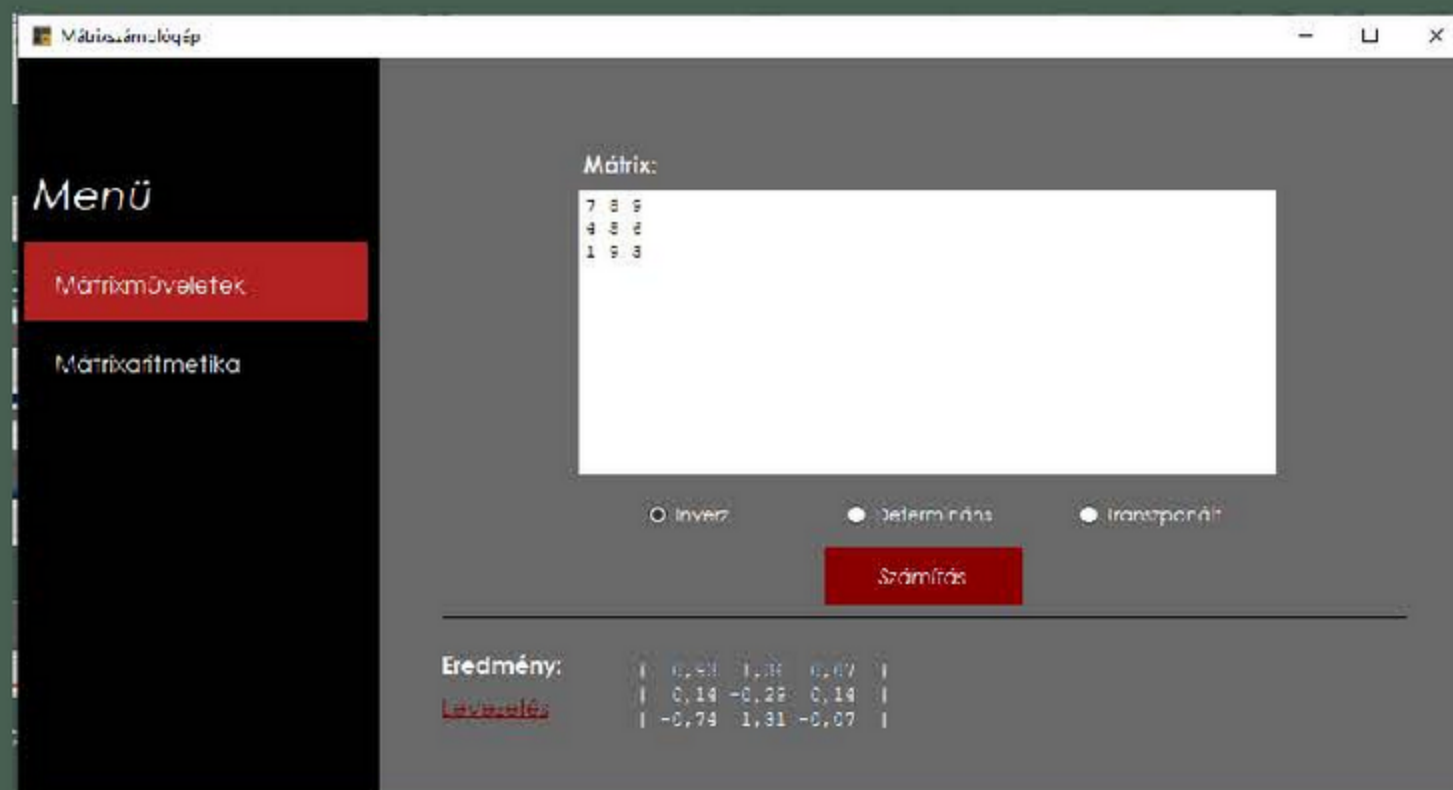
```
void detlevezetes(array<float, 2>^ matrix, int n, int m) {  
    int kul = 0;  
    int hossz = 0;  
    for (int i = 0; i < n; i++) {  
        int maxhossz = 0;  
        Eredmeny->Text += "s" + (i + 1).ToString() + " " + "| ";  
        for (int j = 0; j < m; j++) {  
            for (int p = 0; p < m; p++) {  
                if (matrix[p, j].ToString("F2")->Length > maxhossz) {  
                    maxhossz = matrix[p, j].ToString("F2")->Length;  
                }  
            }  
            hossz = matrix[i, j].ToString("F2")->Length;  
            kul = maxhossz - hossz;  
            for (int l = 0; l < kul; l++) {  
                Eredmeny->Text += " ";  
            }  
            Eredmeny->Text += matrix[i, j].ToString("F2") + " ";  
        }  
        Eredmeny->Text += "|";  
        Eredmeny->Text += "\r\n";  
    }  
}
```

A mátrix pillanatnyi kiírásának kódja

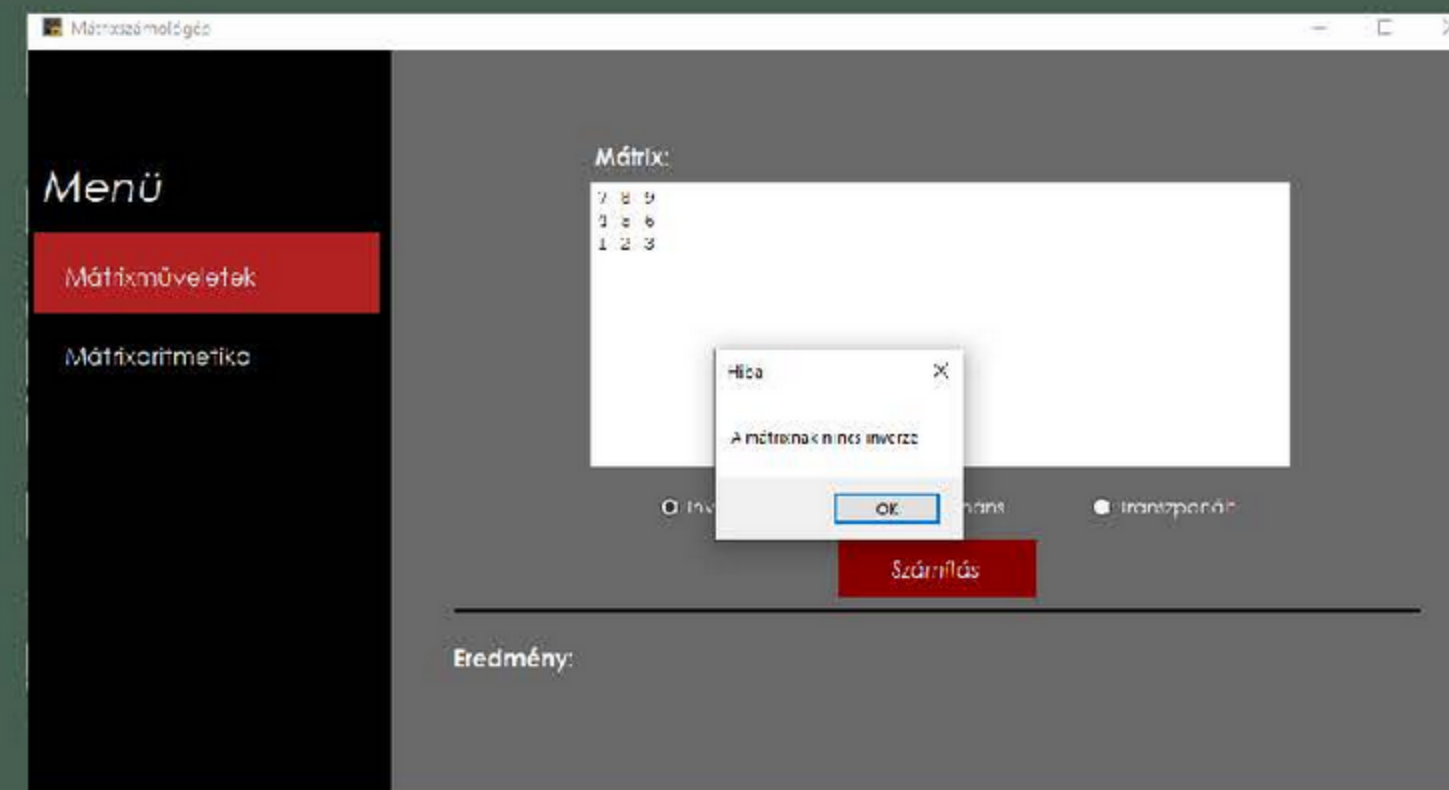
Inverz mátrix

Обернена матриця

- Ha az A $n \times n$ -es mátrixra az $AB = BA = E$ egyenlőség teljesül valamely B mátrix esetén, akkor az A -t invertálható (vagy nemelfajuló) mátrixnak és a B -t az A mátrix inverzének nevezzük.



Inverzmátrix számítás

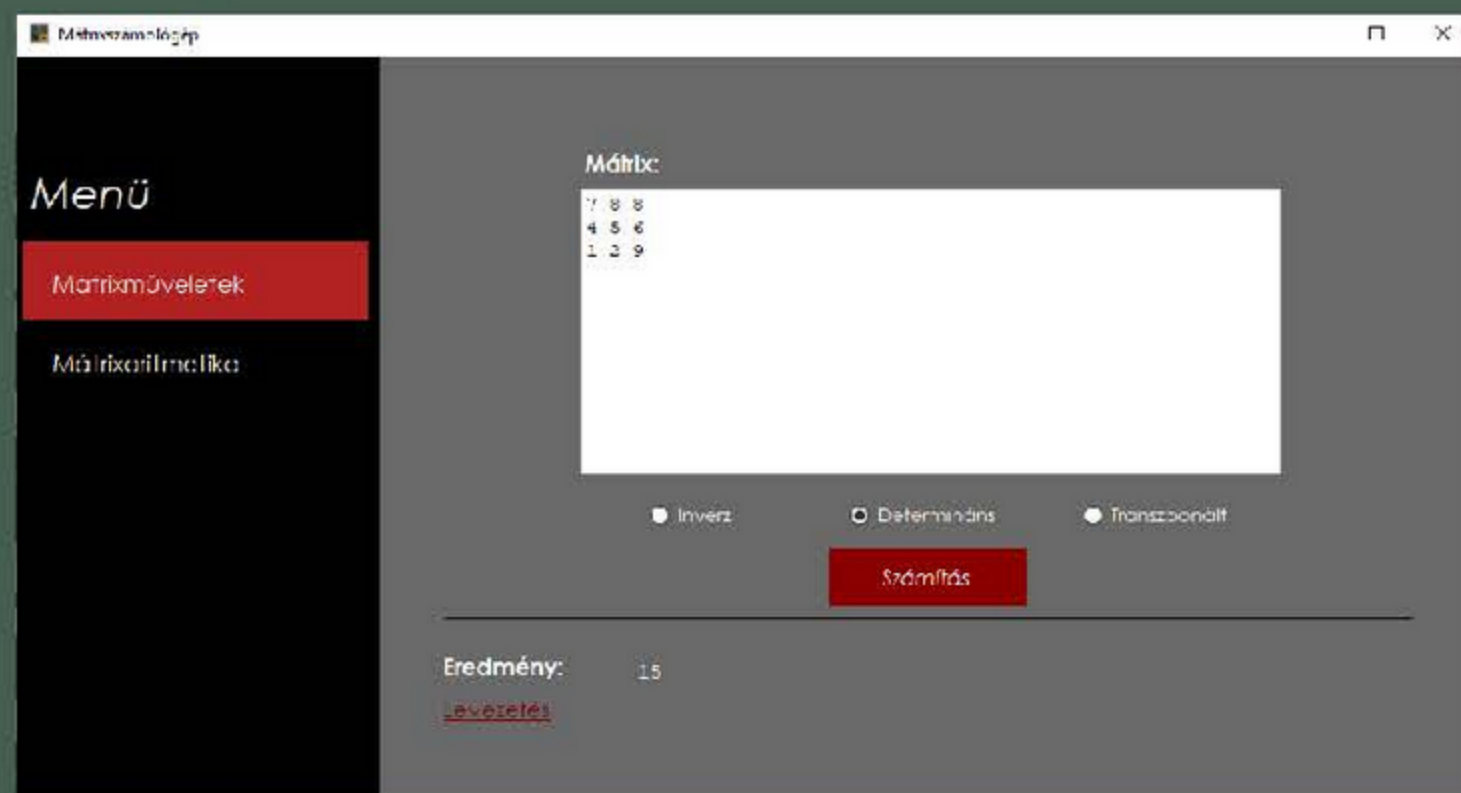


Hibaüzenet

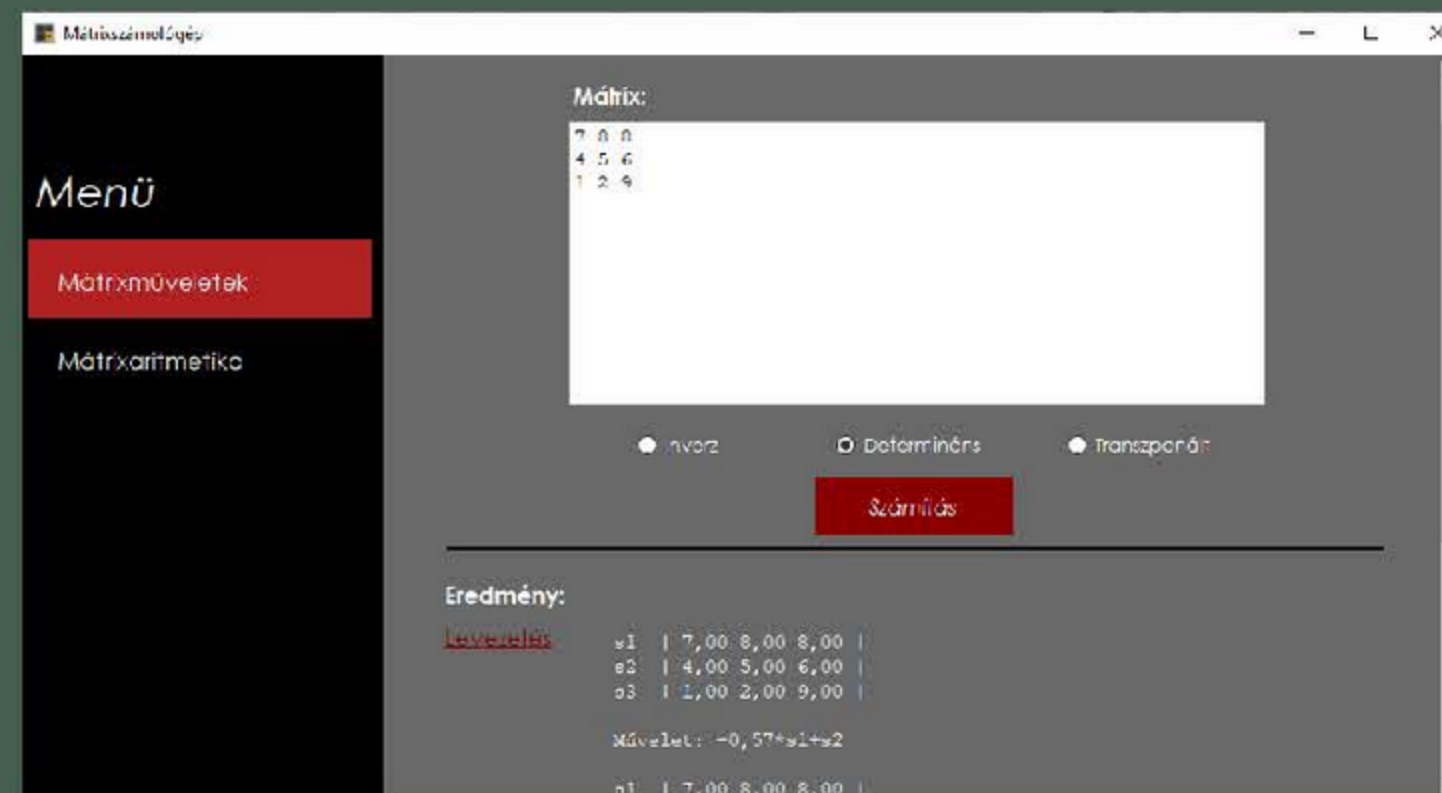
Determináns

Детерминант

- Az A n -es mátrix n -ed rendű determinánsának $n!$ összeadandó algebrai összegét nevezzük, melyben minden tag az A mátrix n elemeinek szorzata, melyeket úgy választunk ki, hogy minden sorból és oszlopból csak egy-egy szerepeljen, emellett az összeadandók előjele az indexekből előálló $\delta \begin{pmatrix} 1 & 2 & 3 \dots n \\ i_1 & i_2 & i_3 \dots i_n \end{pmatrix}$ permutációval határozódik meg a $(-1)^{inv(\delta)}$ hatványból.



Determináns számítás

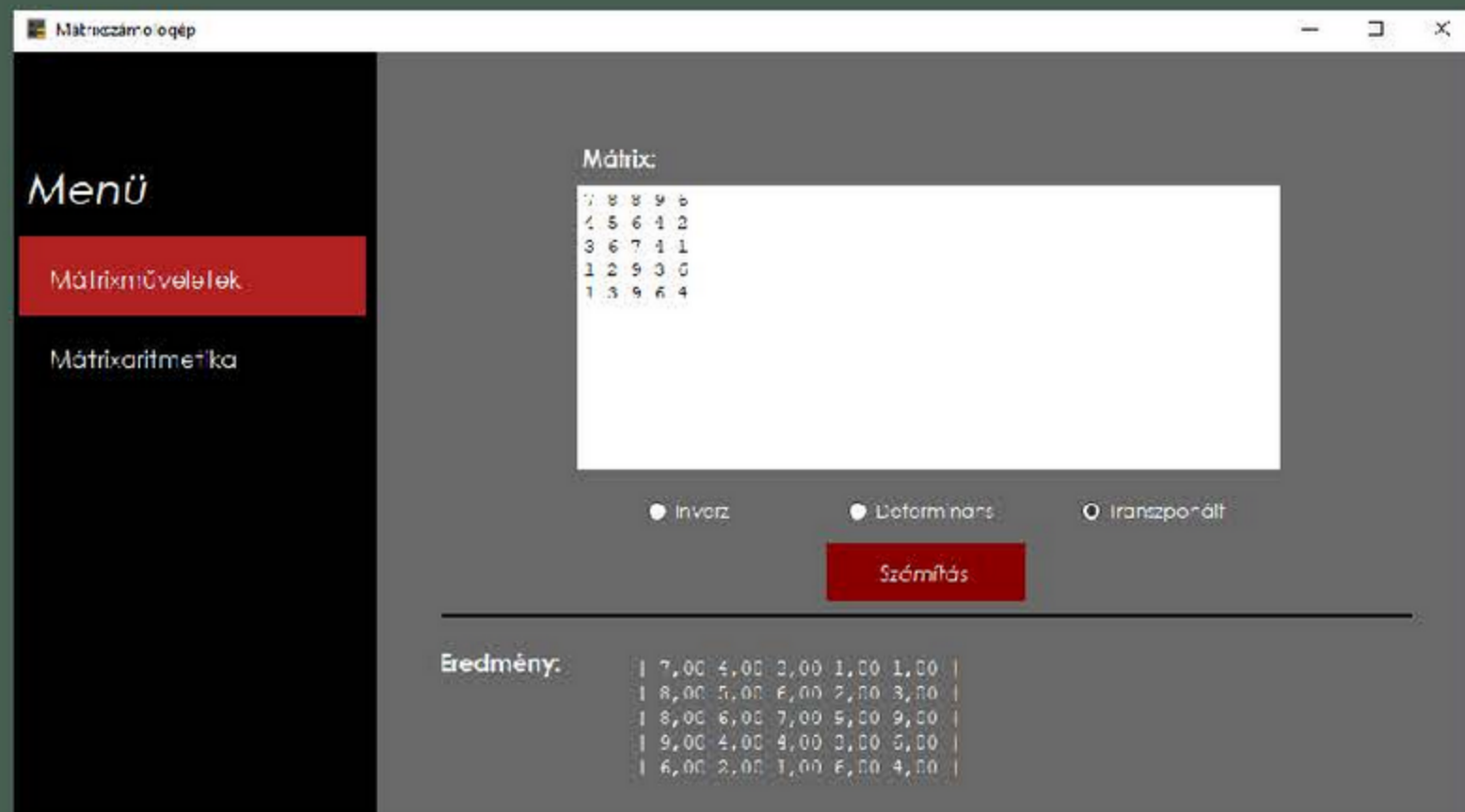


Determináns levezetés

Transzponált mátrix

Транспонированная матрица

- Egy mátrix transzponáltja a sorok és oszlopok felcserélésével nyert mátrix.



Transzponált számítás

```
Transzponálás  
sTranszponalt = "";  
Eredmeny->Visible = false;  
Eredmeny1->Visible = true;  
torol();  
for (int i = 0; i < n1; i++) {  
    int csere;  
    for (int j = i + 1; j < m1; j++) {  
        csere = mm1[i, j];  
        mm1[i, j] = mm1[j, i];  
        mm1[j, i] = csere;  
    }  
}
```

Transzponált számítás kódja

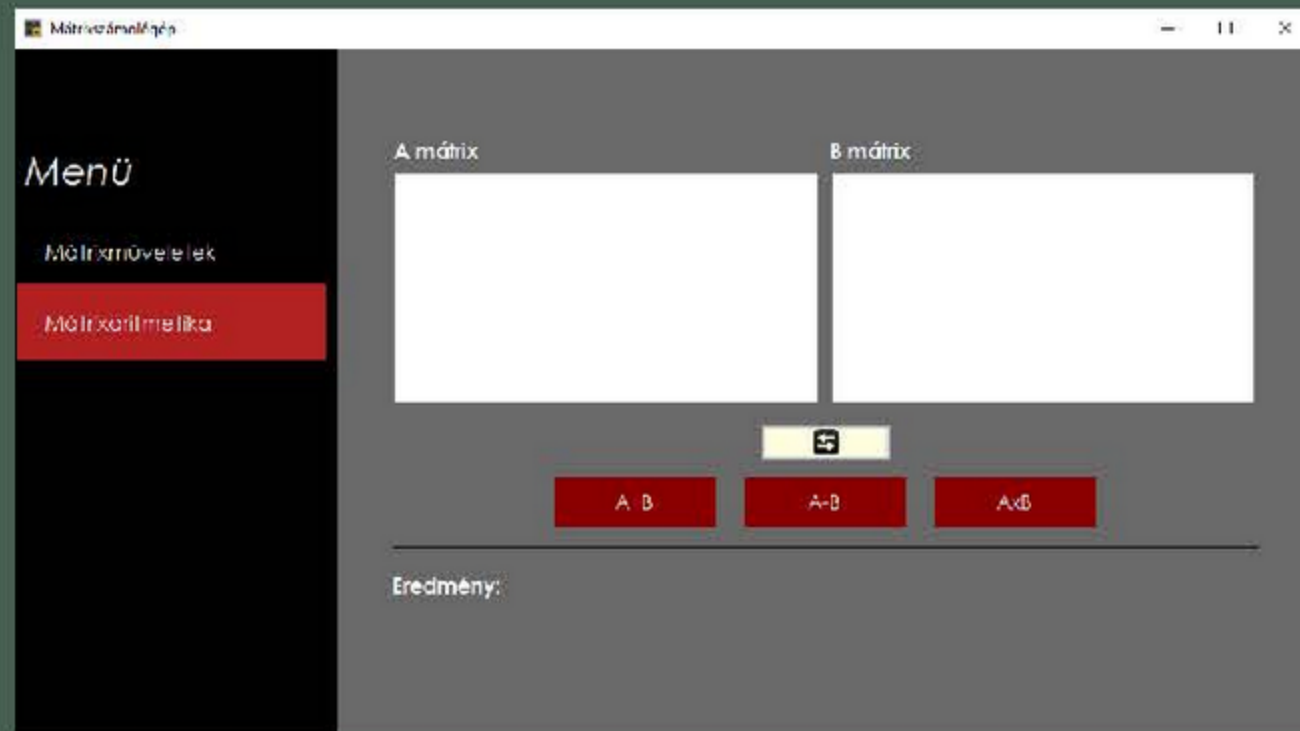
Mátrixaritmetika

Матрична арифметика

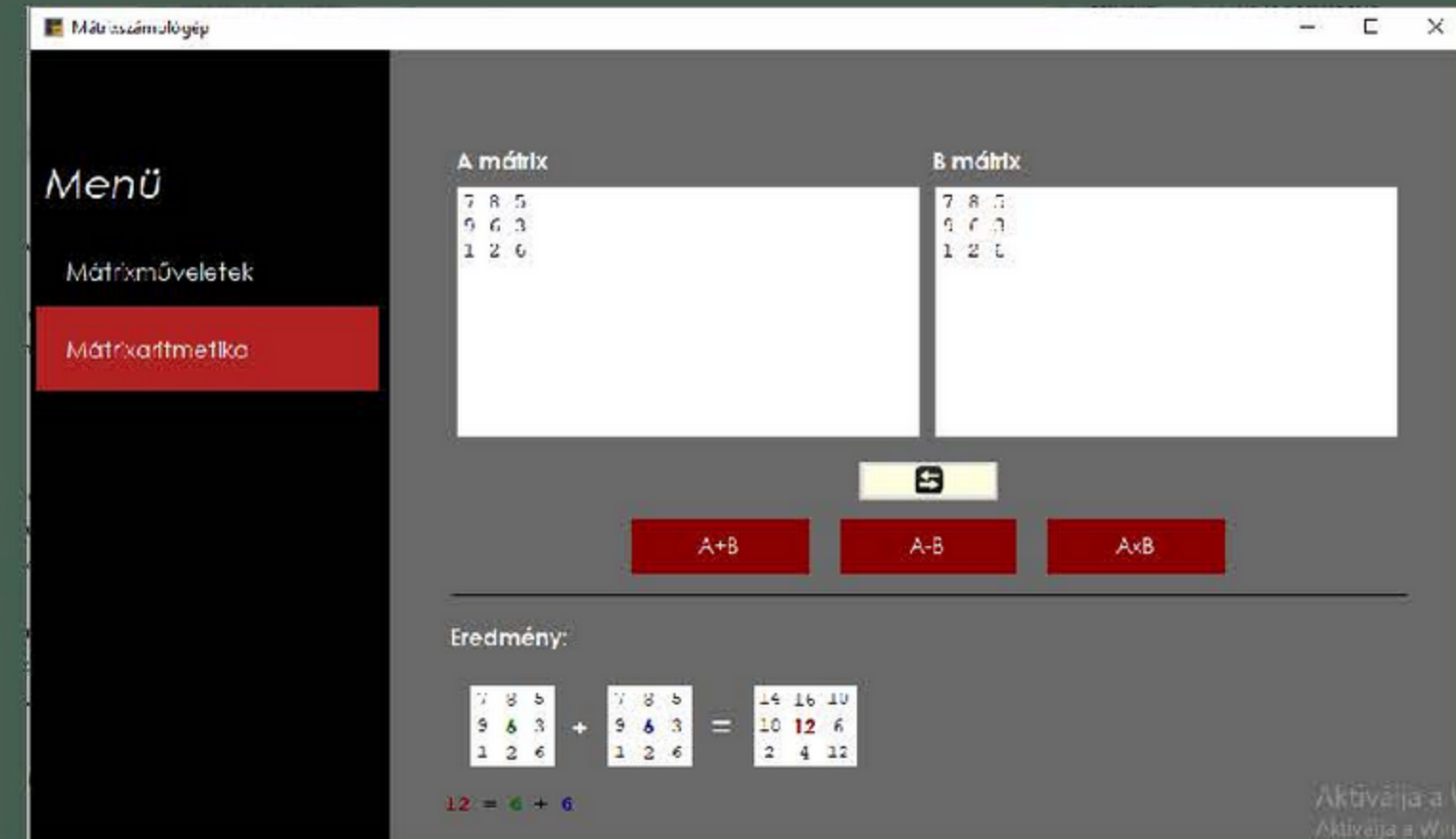
➤ Három művelet tartozik bele:

- Összeadás
- Kivonás
- Szorzás

➤ Műveletek levezetése



Mátrixaritmetika



Eredmény levezetése

Mátrix Összeadás, kivonás

Матричне додавання, віднімання

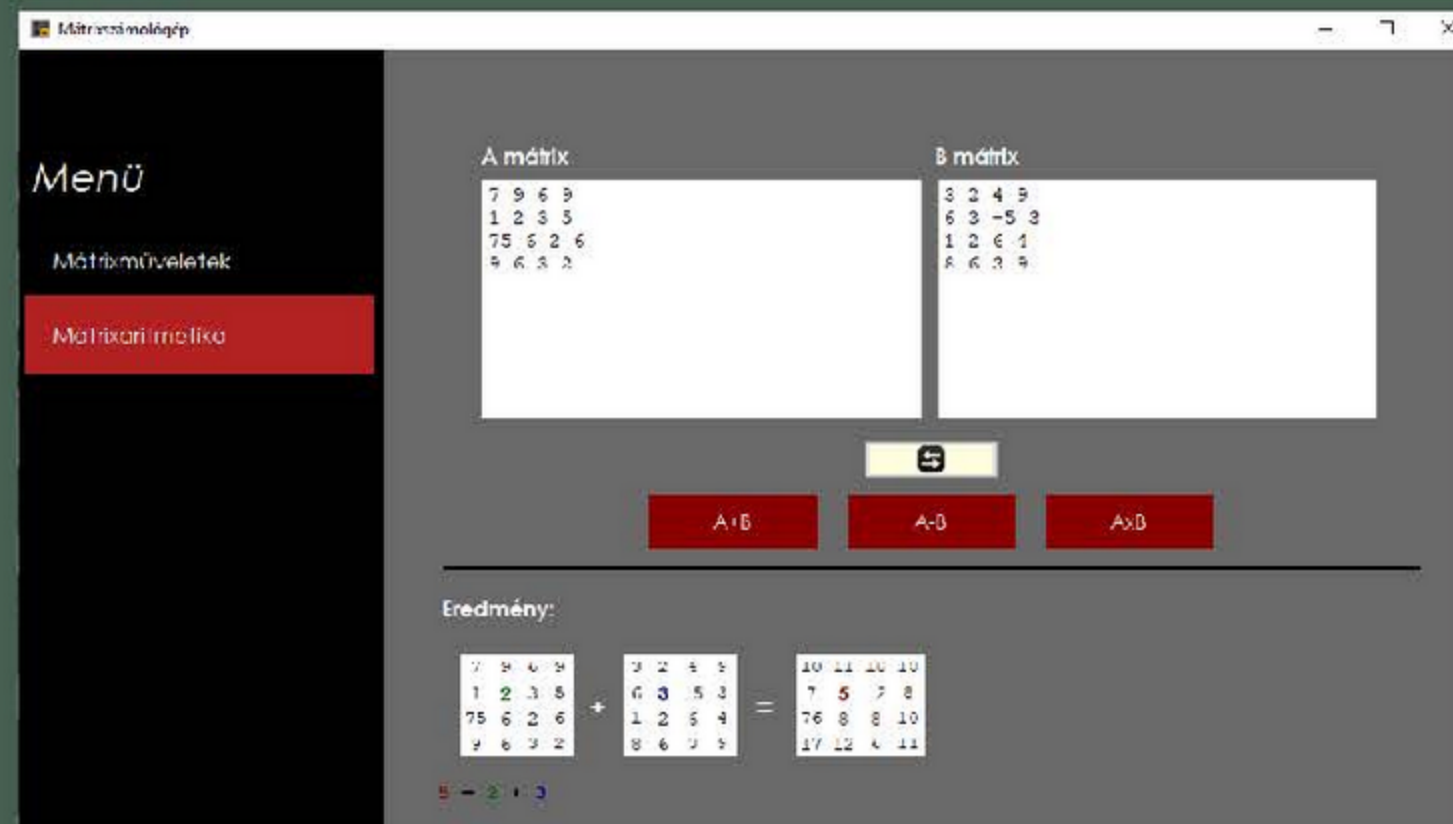
➤ Az $A = (a_{ij})$. és $B = (b_{ij})$ $n \times m$ -es mátrixok $A + B$ összegének azt a $C = (c_{ij})$ mátrixot nevezzük, amelyben $c_{ij} = a_{ij} + b_{ij}$. Azaz a c_{ij} elem az a_{ij} és b_{ij} elemeinek összegével egyenlő.

```
array<double, 2>^ osszeg = gcnew array<double, 2>(n2, m2);
String^ sOsszeg = "";
for (int i = 0; i < n1; i++) {
    for (int j = 0; j < m1; j++) {
        osszeg[i, j] = mm1[i, j] + mm2[i, j];
        sOsszeg += Convert::ToString(osszeg[i, j]) + " ";
    }
    sOsszeg += "\r\n";
}
eredmeny->ColumnCount = m2;
aMatrix->ColumnCount = m1;
bMatrix->ColumnCount = m2;

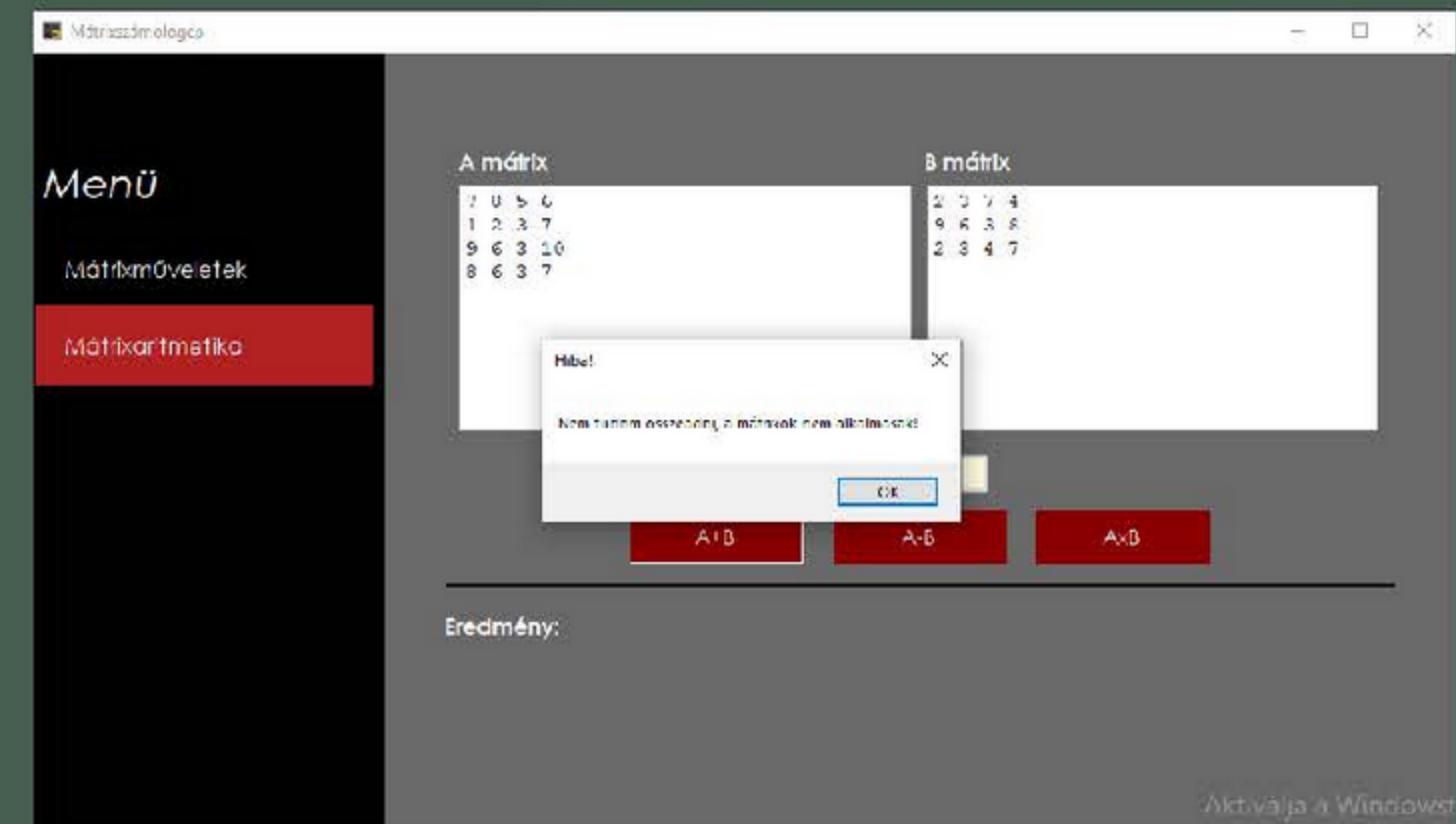
array<String^>^ eSor = gcnew array<String^>(m2);
array<String^>^ aSor = gcnew array<String^>(m1);
array<String^>^ bSor = gcnew array<String^>(m2);

for (int i = 0; i < n1; i++) {
    for (int j = 0; j < m1; j++) {
        aSor[j] = mm1[i, j].ToString();
    }
    aMatrix->Rows->Add(aSor);
}
```

Az összeadás számítás kódja



Összeg számítás és levezetés



Hibaüzenet

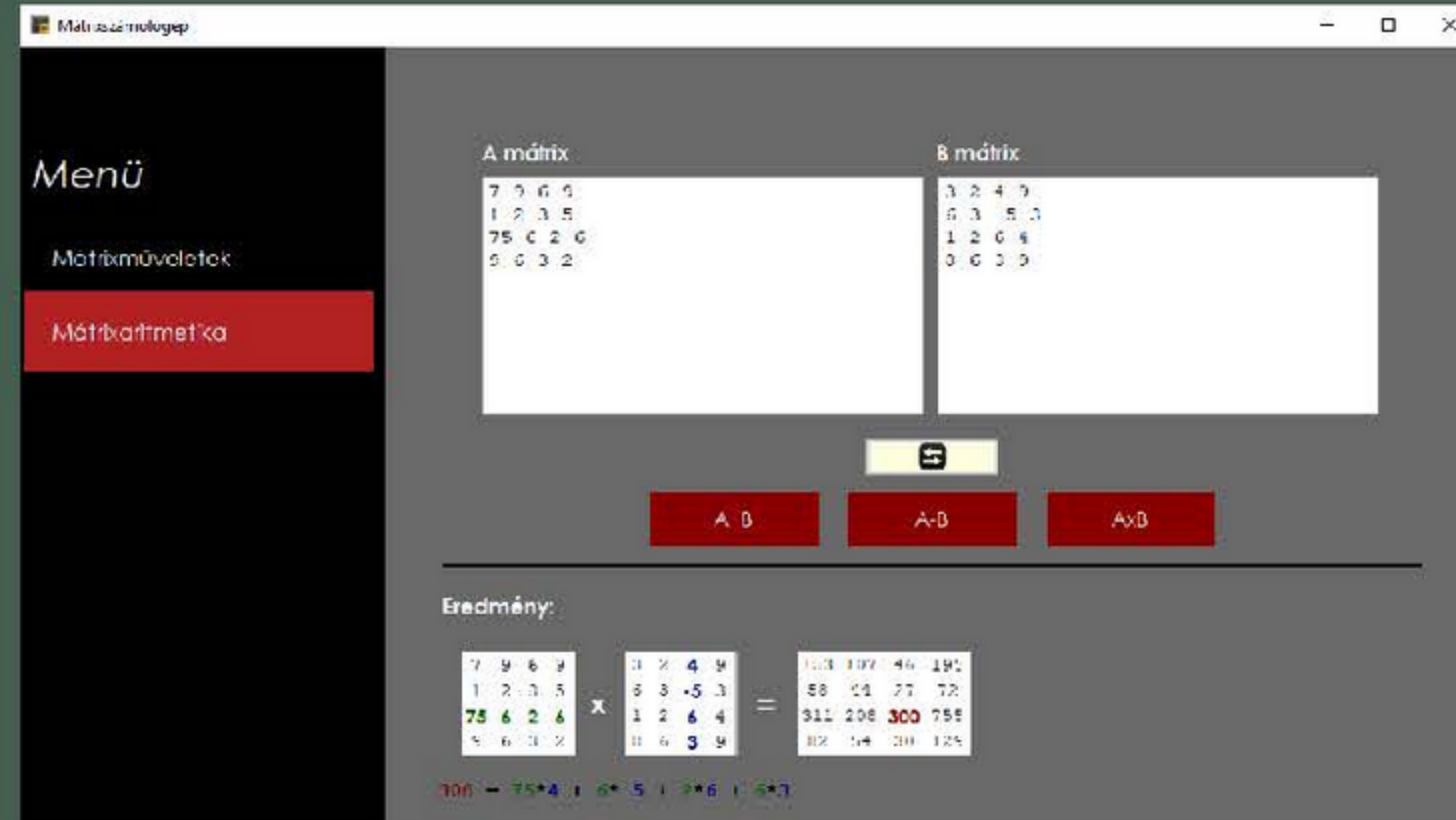
Mátrixok szorzása

Матричное множение

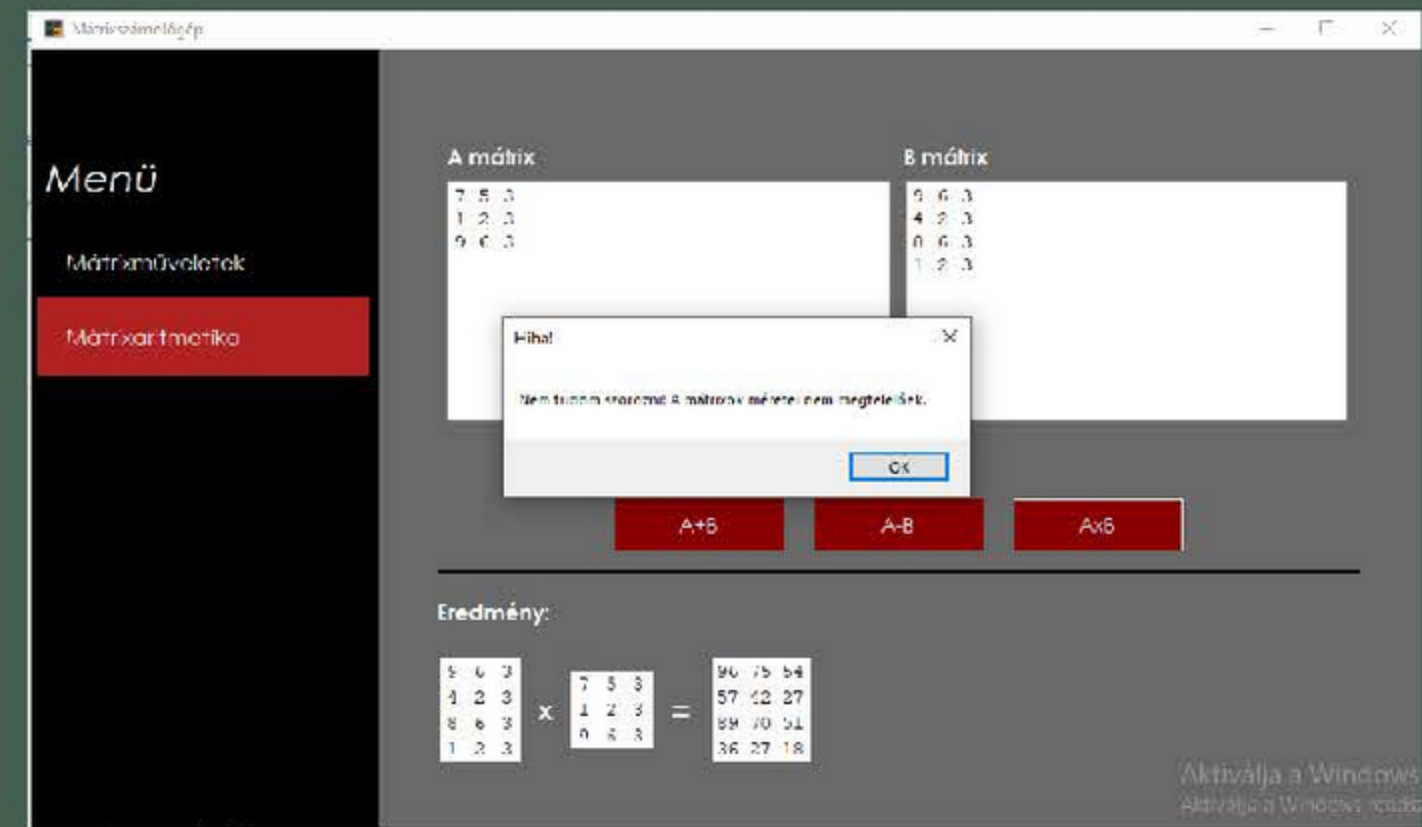
- Az $A = (a_{ij})$ és $B = (b_{ij})$ mátrixok AB szorzatának azt a $C = (c_{ij})$ mátrixot értjük, melyben mátrix c_{ij} -ik eleme az A mátrix i-ik sorában és a mátrix j-ik oszlopában lévő megfelelő elemek szorzatainak összege.

```
array<double, 2>^ szorzat = gcnew array<double, 2>(n1, m2);
for (int i = 0; i < n1; i++) {
    for (int j = 0; j < m2; j++) {
        szorzat[i, j] = 0;
        for (int k = 0; k < m2; k++) {
            szorzat[i, j] += mm1[i, k] * mm2[k, j];
        }
    }
}
eredmeny->ColumnCount = m2;
aMatrix->ColumnCount = m1;
bMatrix->ColumnCount = m2;

array<String^>^ eSor = gcnew array<String^>(m2);
```



Szorzat számítás



Hibaüzenet

Numerikus integrálási módszereket bemutató alkalmazás fejlesztése

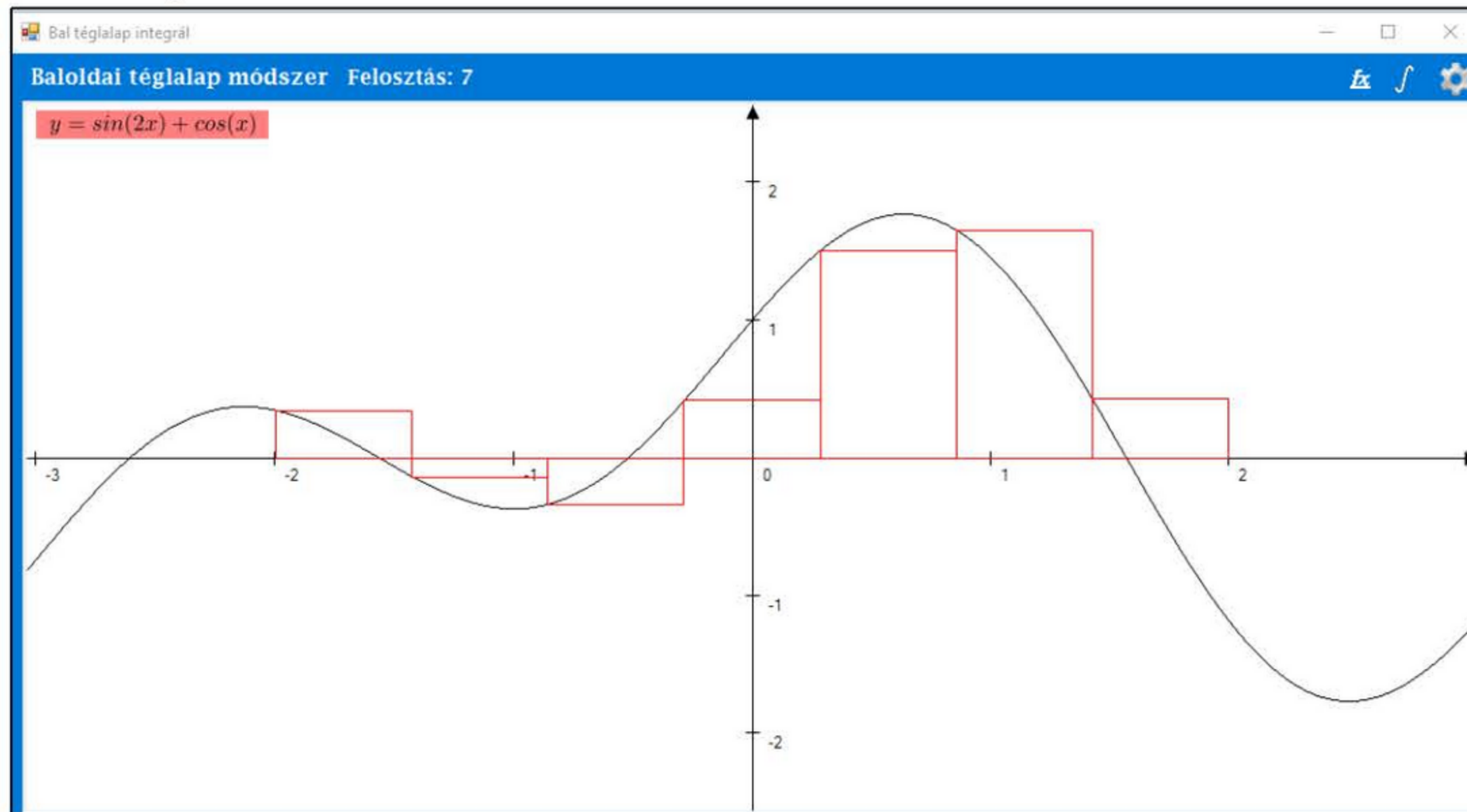
Розробка програмного засобу для презентації методів
чисельного інтегрування функцій

Készítette: Szabó Bettina,
Szak: Alkalmazott matematika III/6.
Vezető tanár: Szilágyi Lajos

Baloldali téglalap módszer

Метод лівого прямокутника

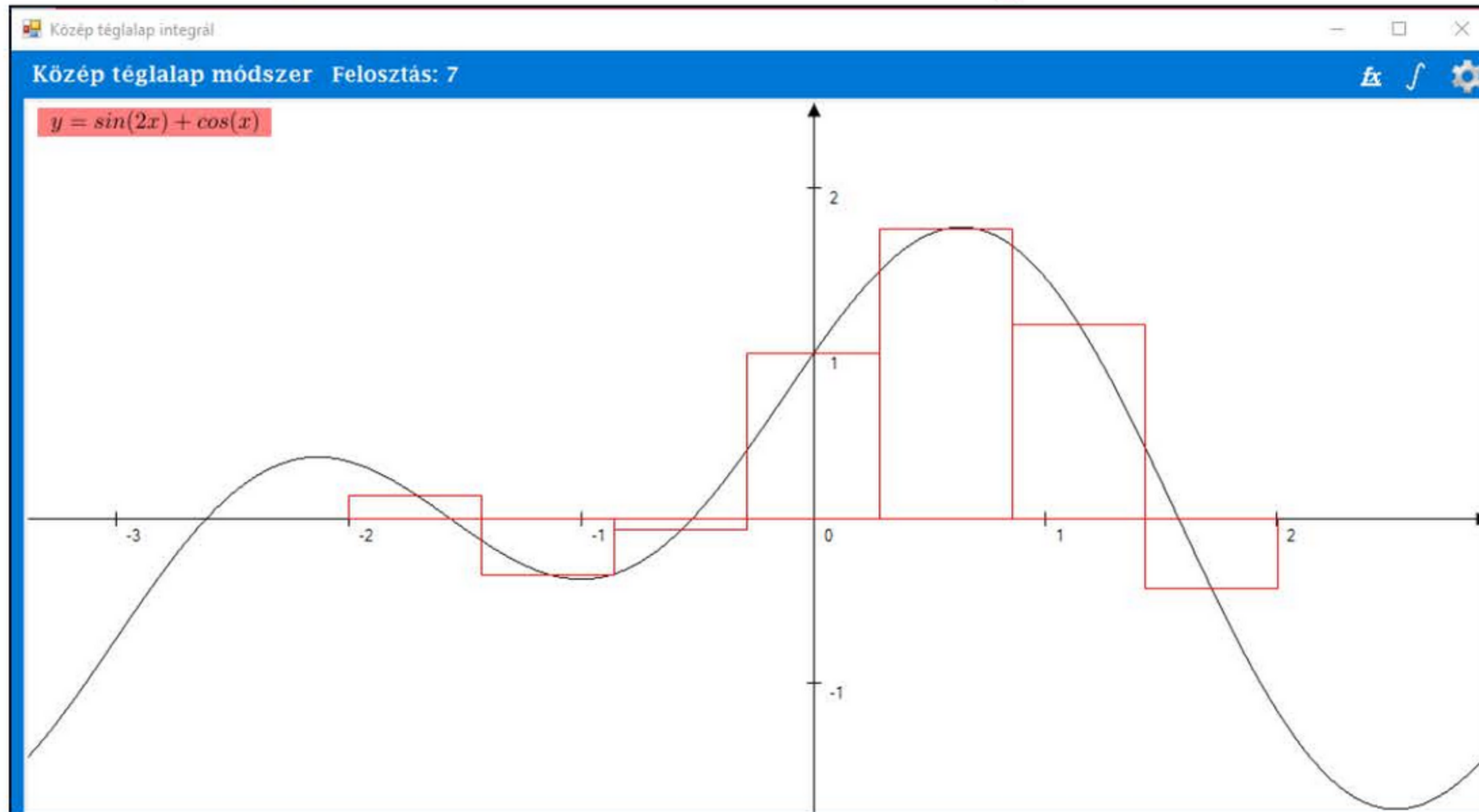
Olyan *téglalapokkal* közelíti az f függvénygörbe alatti területet, melyek érintik a függvénygörbét a részintervallumok kezdő értékeiben. $\int_a^b f(x)dx = h * \sum_{i=0}^{n-1} f(x_i)$



Közép téglalap módszer

Метод середнього прямокутника

Téglalapokkal közelíti az f függvénygörbe alatti területet. A függvény értékét a részintervallumok felezőpontjában kell meghatározni. $\int_a^b f(x)dx = h * \sum_{i=0}^{n-1} f(\frac{x_i+x_{i+1}}{2})$

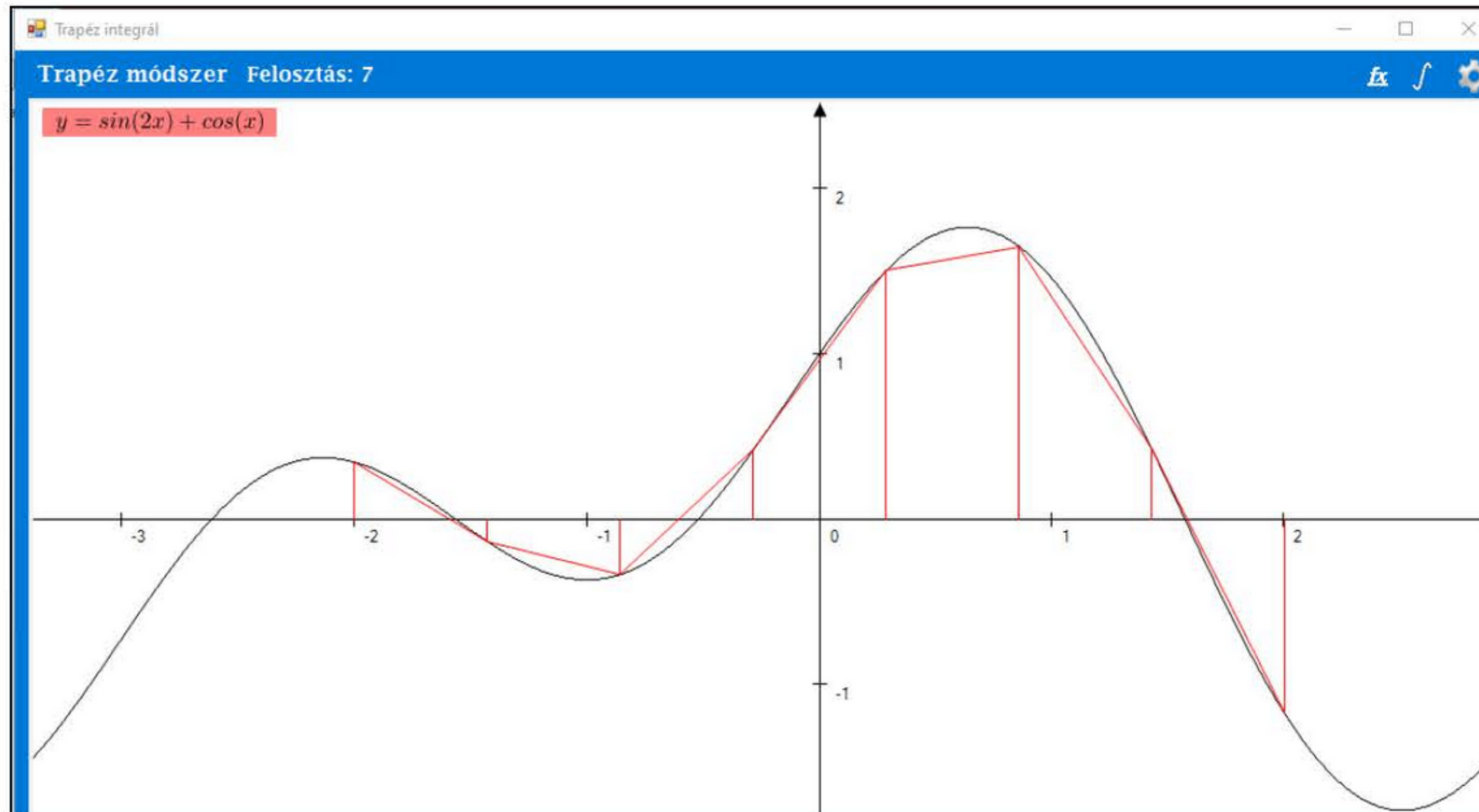


Trapéz módszer

Трапеційний метод

Trapézokkal közelíti az f függvénygörbe alatti területet.

$$\int_a^b f(x)dx \approx \frac{h}{2} * (f(x_0) + f(x_n) + 2 * \sum_{i=1}^{n-1} f(x_i))$$

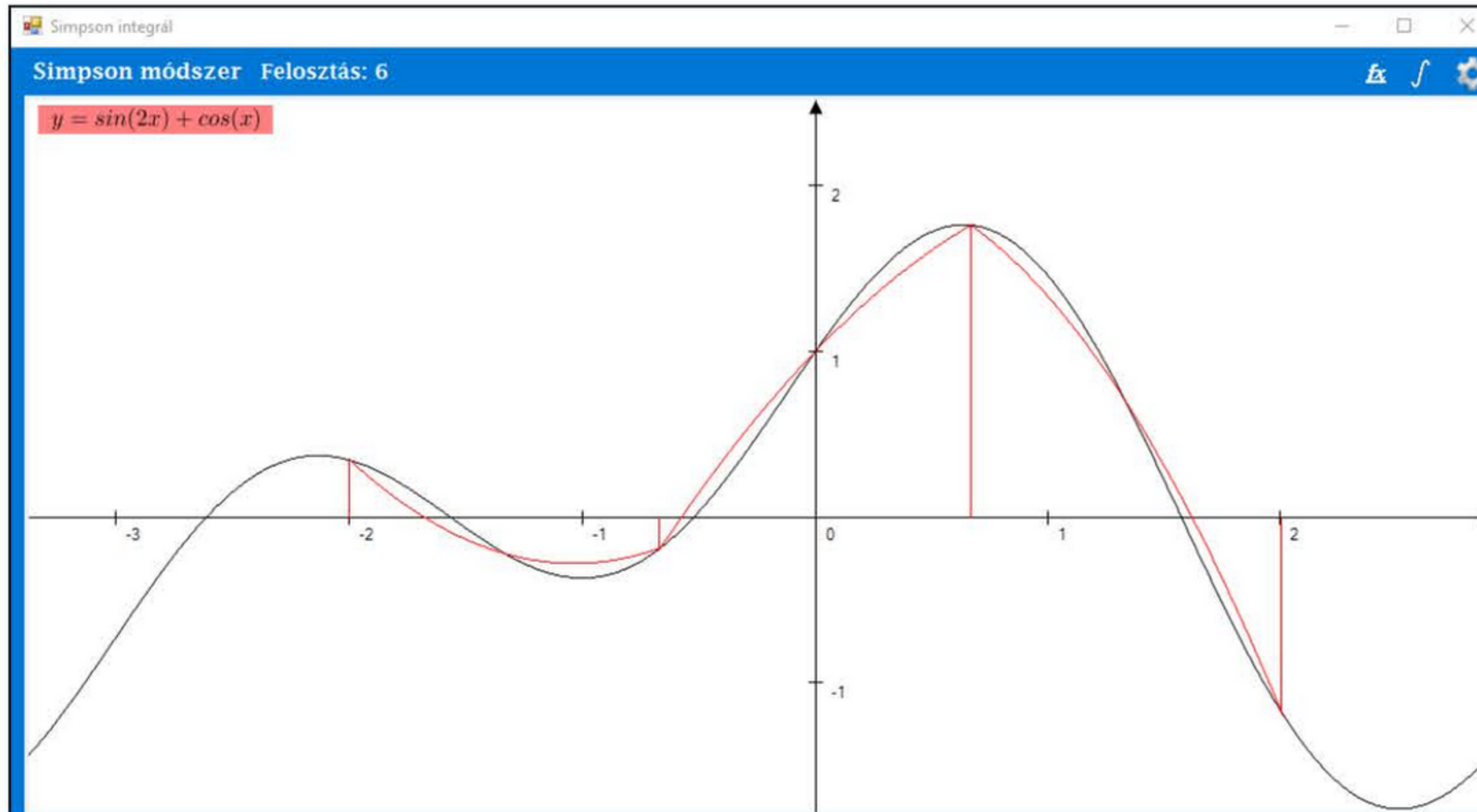


Simpson módszer

Метод Симпсона

Parabolákkal közelíti az f függvénygörbe alatti területet. Páros számú felosztásra alkalmazható a módszer. A parabolákat

$$\int_a^b f(x)dx \approx \frac{h}{3} * (f(x_0) + f(x_n) + 2 * \sum_{i=2,4,..}^{n-2} f(x_i) + 4 * \sum_{i=1,3}^{n-1} f(x_i))$$



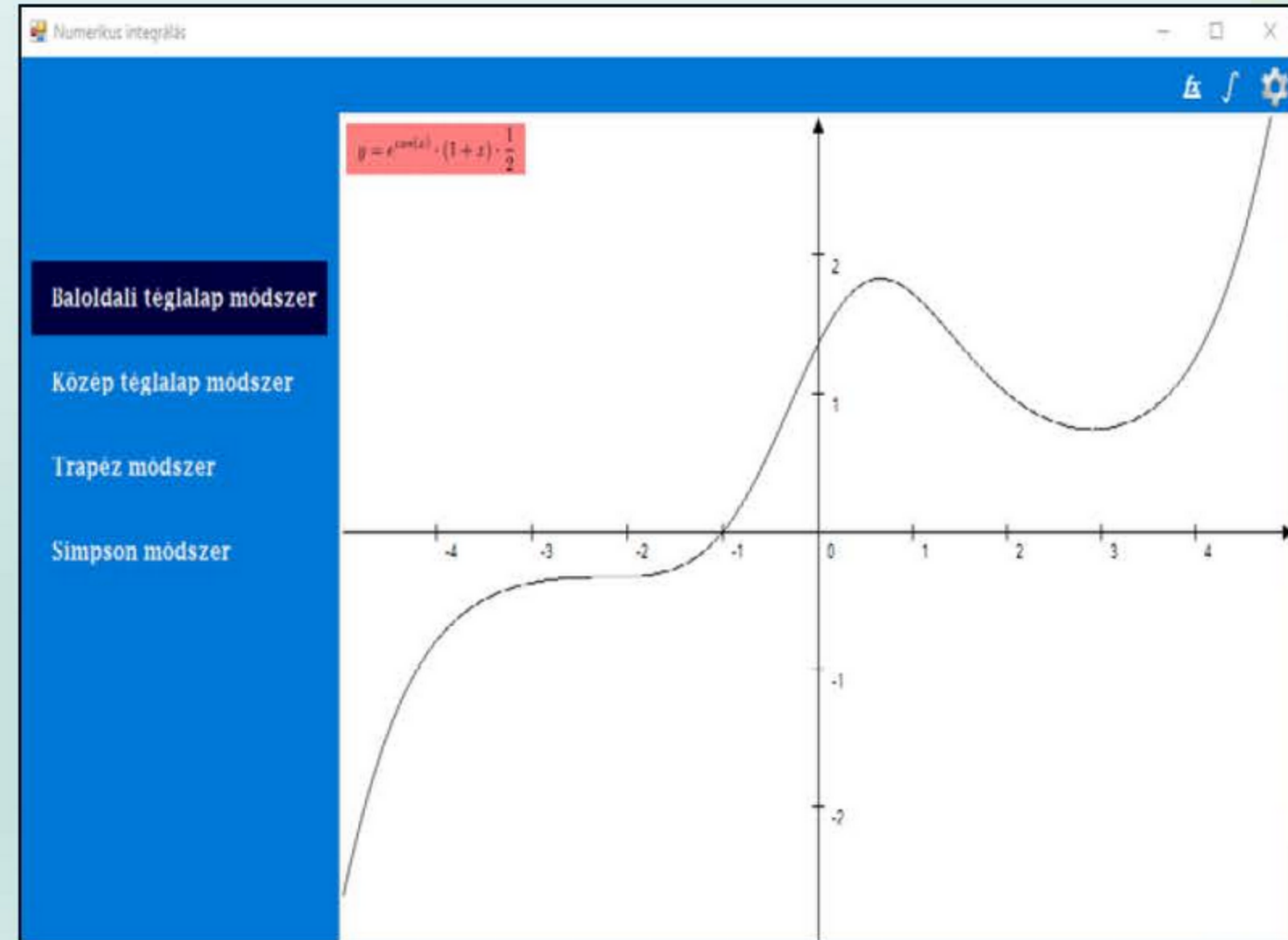
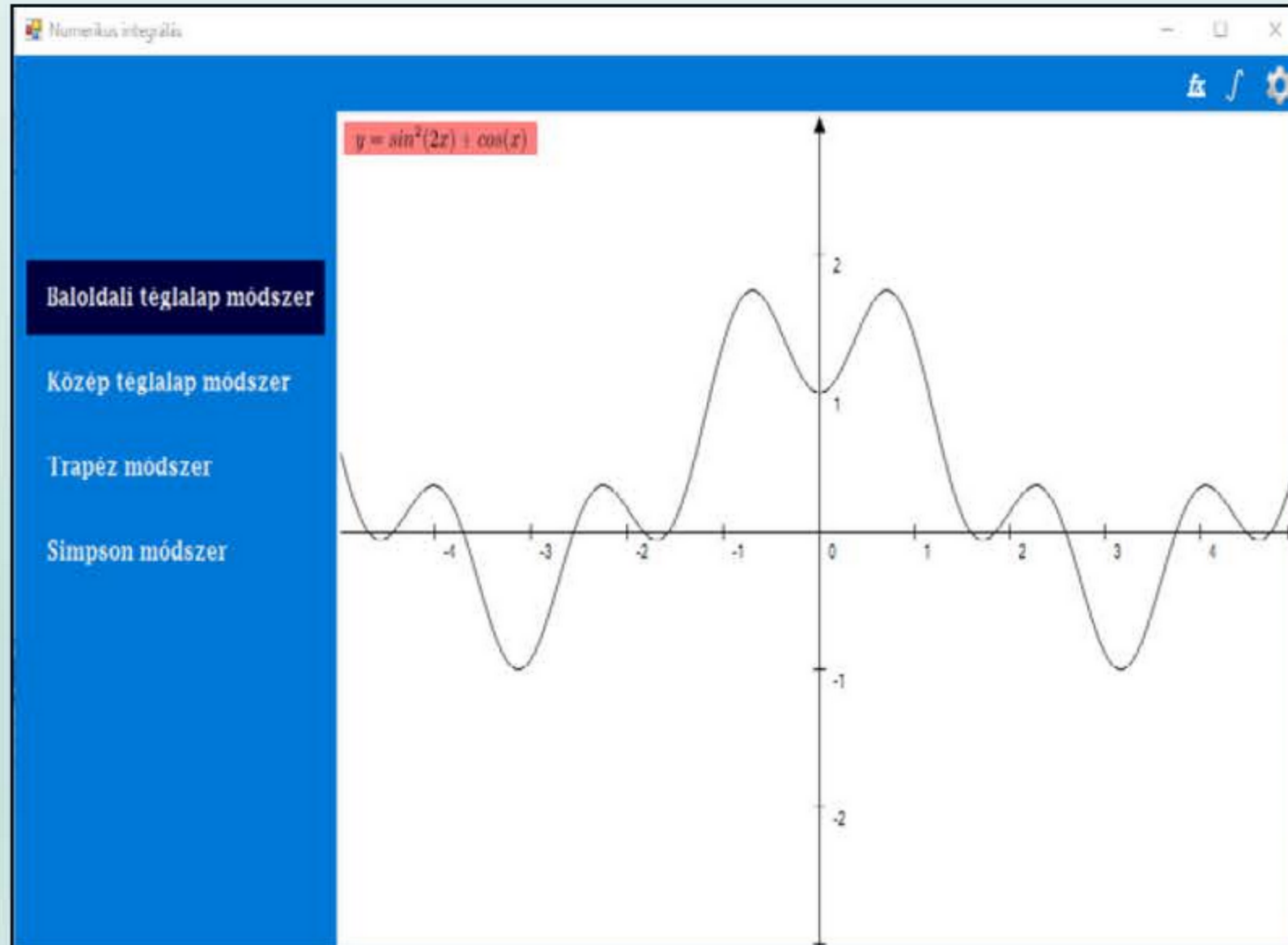
Az alkalmazás

Програмне забезпечення

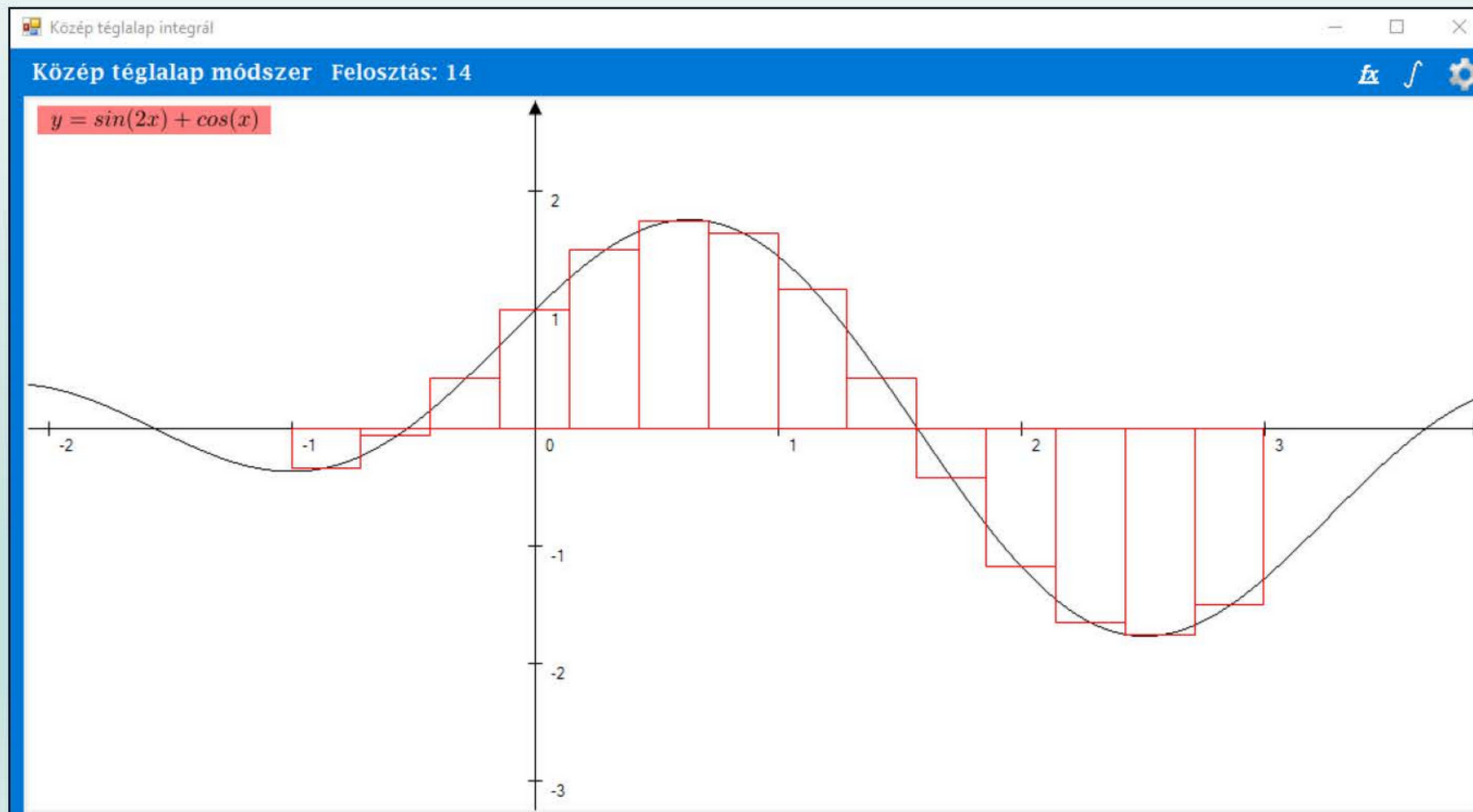
Különböző numerikus integrálási módszerek grafikusan szemléltetése.

Az alkalmazással megvalósítható:

- A koordináta tengelyek és a függvények grafikonjainak megrajzoltatása.

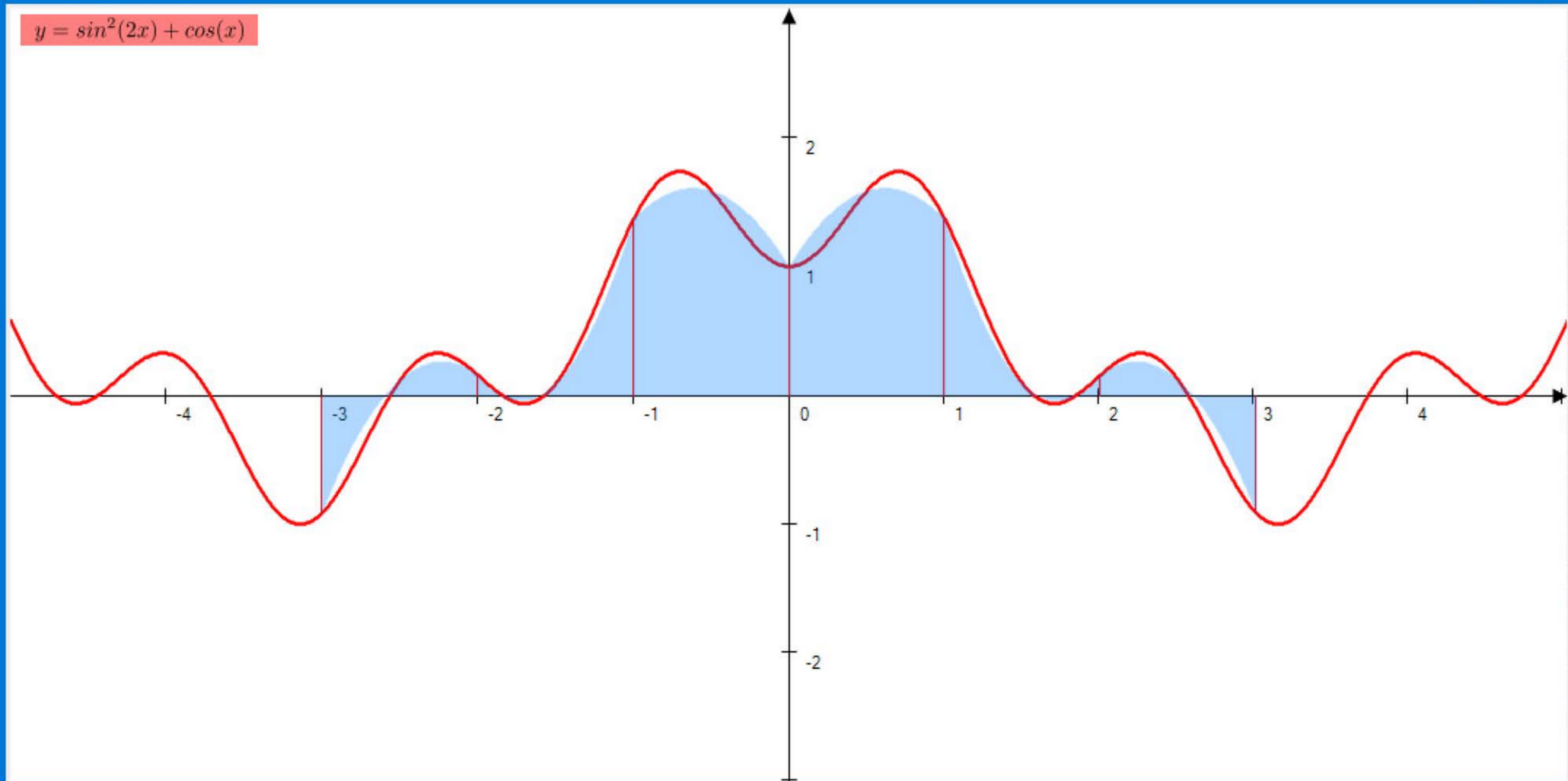


- A különböző numerikus integrálási módszerek menetének szemléltetése.
- Ілюстрація процесу різних методів чисельного інтегрування.

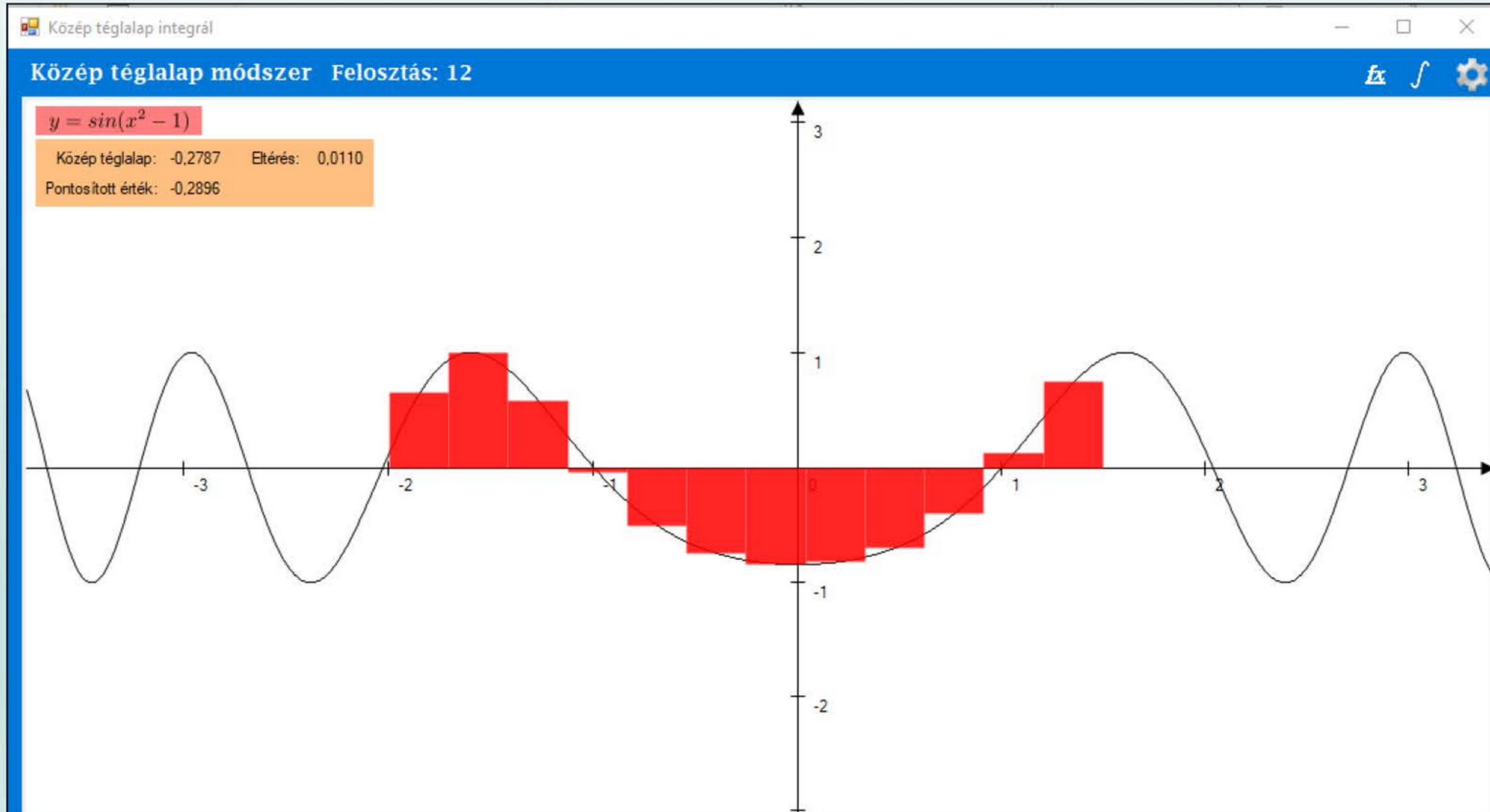


Simpson módszer Felosztás: 12

$$y = \sin^2(2x) + \cos(x)$$



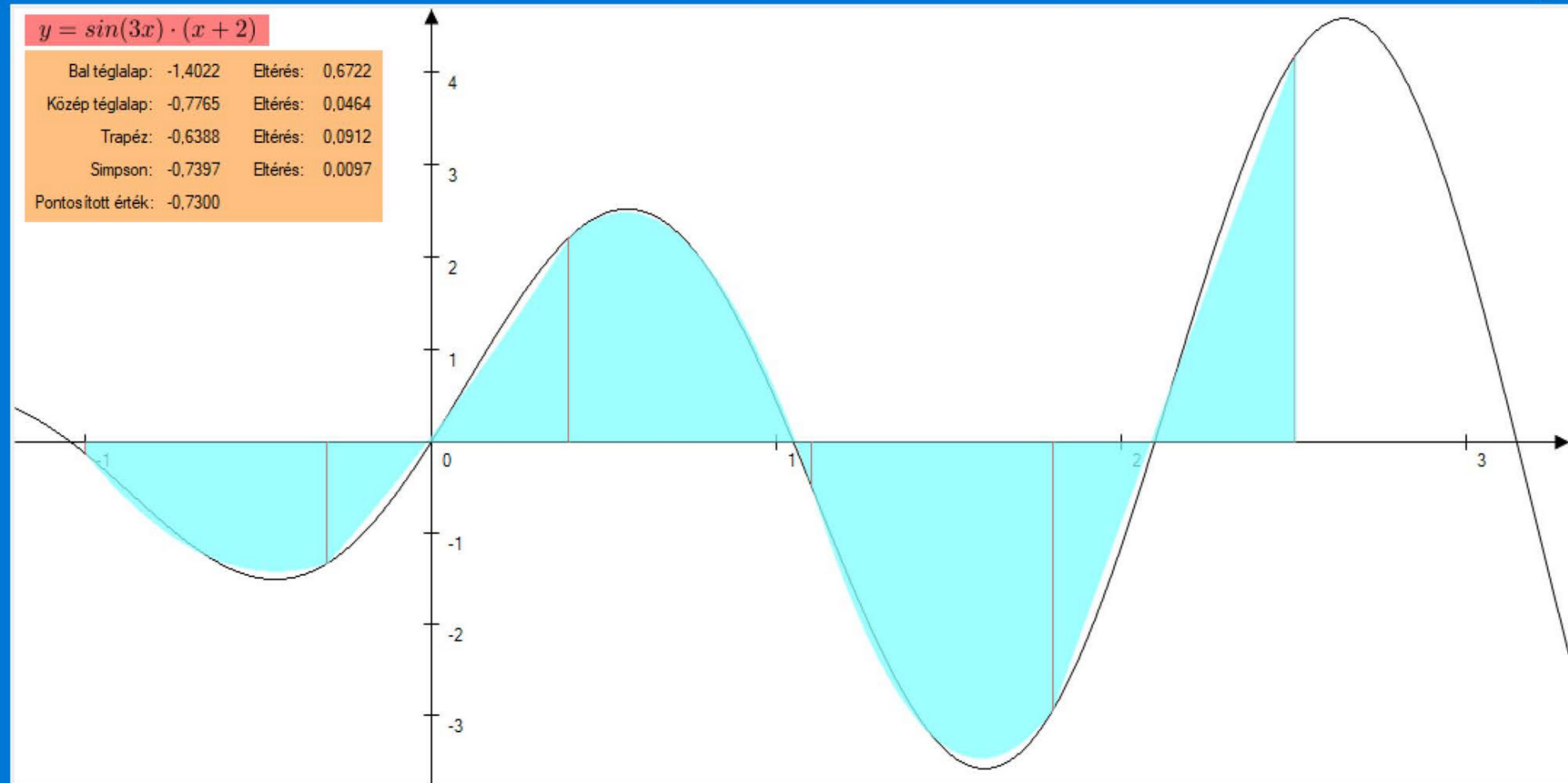
- A kapott értékek és azoknak a pontosított értéktől való eltéréseinek megjelenítése.



Simpson módszer Felosztás: 10

$$y = \sin(3x) \cdot (x + 2)$$

Bal téglalap:	-1,4022	Eltérés:	0,6722
Közép téglalap:	-0,7765	Eltérés:	0,0464
Trapéz:	-0,6388	Eltérés:	0,0912
Simpson:	-0,7397	Eltérés:	0,0097
Pontosított érték:	-0,7300		



**HŐMÉRSÉKLET-ÉRZÉKELŐ ÉS FIGYELŐ RENDSZER
LÉTREHOZÁSA ÉS BEPROGRAMOZÁSA RASBIAN OPERÁCIÓS
RENDSZER ALATT RASPBERRY PI PLATFORMON**

**СТВОРЕННЯ ТА ПРОГРАМУВАННЯ ДАТЧИКА ТЕМПЕРАТУРИ
ТА СИСТЕМИ МОНІТОРИНГУ В ОПЕРАЦІЙНІЙ СИСТЕМІ
RASPBIAN НА ПЛАТФОРМІ RASPBERRY PI**

Készítette: Tihor Fruzsina
AM-III/6

Témavezető: Makó Pál

Hőmérséklet mérés

Вимірювання температури



↑
DS1820 hőérzékelő



↑
Hőérzékelők csatlakoztatása a Raspberry Pi-hez

Thonny - /home/pi/Desktop/FRUZZSINA_02032020/homerseklat/3_sensor_homero_C_K_0203.py @ 42:23

NewLoadSaveRunDebugOverInfoOutStopZoomQuit

Switch to regular mode

vegso.proba.py 3_sensor_homero_C_K_0203.py

```
31 if equals_pos[0] != -1 and equals_pos[1] != -1 and equals_pos[2] != -1:
32     temp_string = [lines[0][1][equals_pos[0]+2:], lines[1][1][equals_pos[1]+2:], lines[2]
33     temp_c= [0,0,0]
34     temp_k=[0,0,0]
35     for i in range(0,3):
36         temp_c[i] = float(temp_string[i]) / 1000.0
37         temp_k[i] = float(temp_c[i]) + 273
38     print ("Homero 1")
39     print(temp_c[0], "°C", temp_k[0], "K")
40     print ("Homero 2")
41     print(temp_c[1], "°C", temp_k[1], "K")
42     print ("Homero 3")
43     print(temp_c[2], "°C", temp_k[2], "K")
44     print("\n")
45
46
47 while True:
48     read_temp()
49     time.sleep(1)
```

Shell

```
>>> %Run 3_sensor_homero_C_K_0203.py
Homero 1
19.25 °C 292.25 K
Homero 2
18.937 °C 291.937 K
Homero 3
19.062 °C 292.062 K
```


Páratartalom mérés Вимірювання вологості



DHT22 páratartalomérzékelő



Páratartalomérzékelők csatlakoztatása a Raspberry Pi-hez

Thonny - /home/pi/Desktop/FRUZZSINA_02032020/paratartalom/Harmatpont_keplet_02032020.py @ 16:45

NewLoadSaveRunDebugOverIntoOutStopZoomQuit

Switch to regular mode

vegso.proba.py 3_sensor_homero_C_K_0203.py Harmatpont_keplet_02032020.py

```
1
2 import Adafruit_DHT as dht
3 from time import sleep
4 import math
5
6 DHT = 10
7 while True:
8
9     h,t = dht.read_retry(dht.DHT22, DHT)
10
11     print('Hőmérséklet={0:0.2f}°C Páratartalom={1:0.2f}%'.format(t,h))
12     Stg = 0.021*t*t+0.1967*t+5.1625
13     c = 5.1625-(h*Stg)/100
14     D = 0.1967 * 0.1967 - 4 * 0.021 * c
15     x = (-0.1967 + math.sqrt(D)) / (2 * 0.021)
16     print('Harmatpont={0:0.2f}°C'.format(x))
17     sleep(1)
```

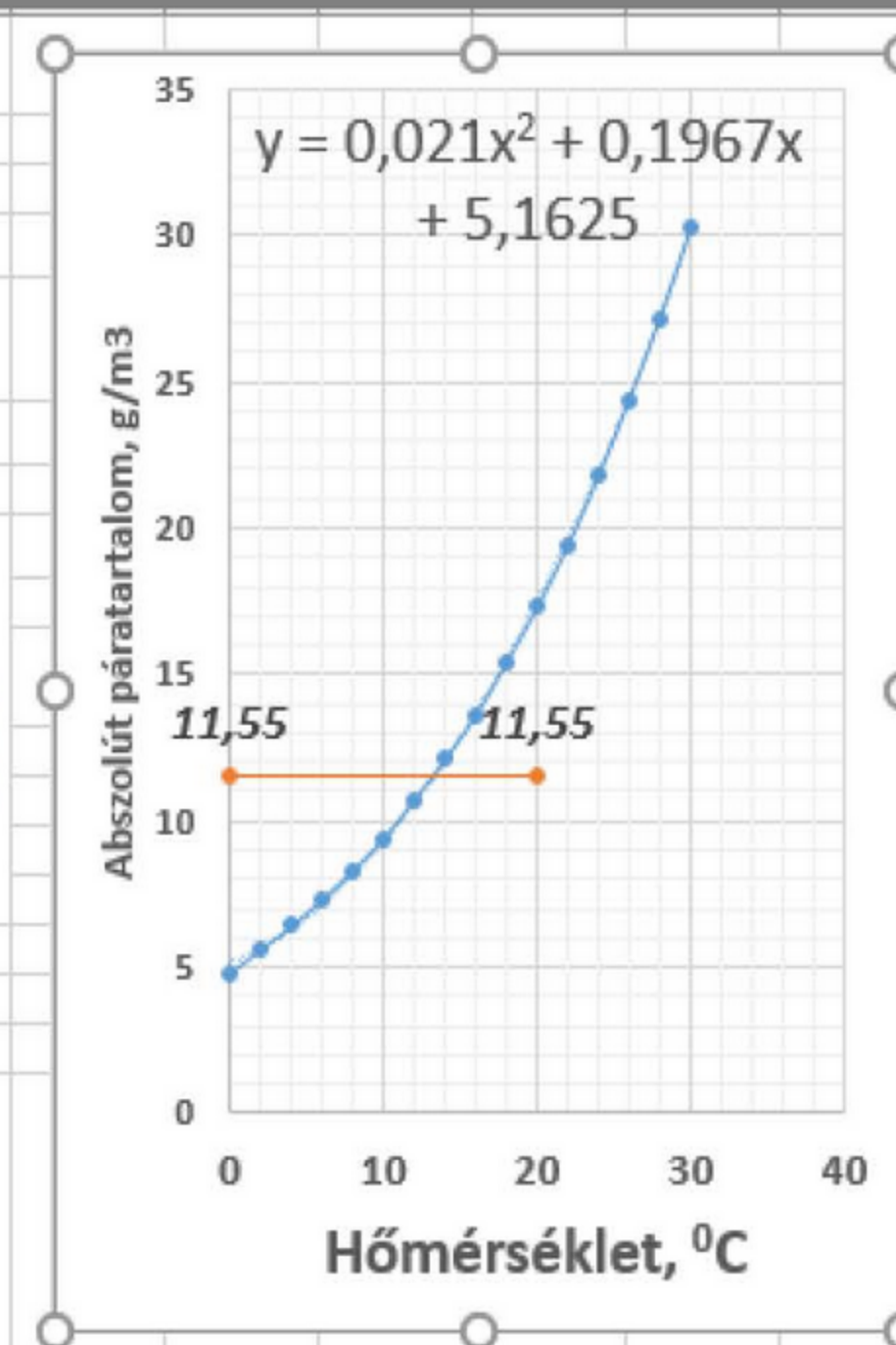
Shell

```
>>> %Run Harmatpont_keplet_02032020.py
Hőmérséklet=20.70°C Páratartalom=48.30%
Harmatpont=9.30°C
Hőmérséklet=20.70°C Páratartalom=48.50%
Harmatpont=9.36°C
Hőmérséklet=20.70°C Páratartalom=48.50%
Harmatpont=9.36°C
```

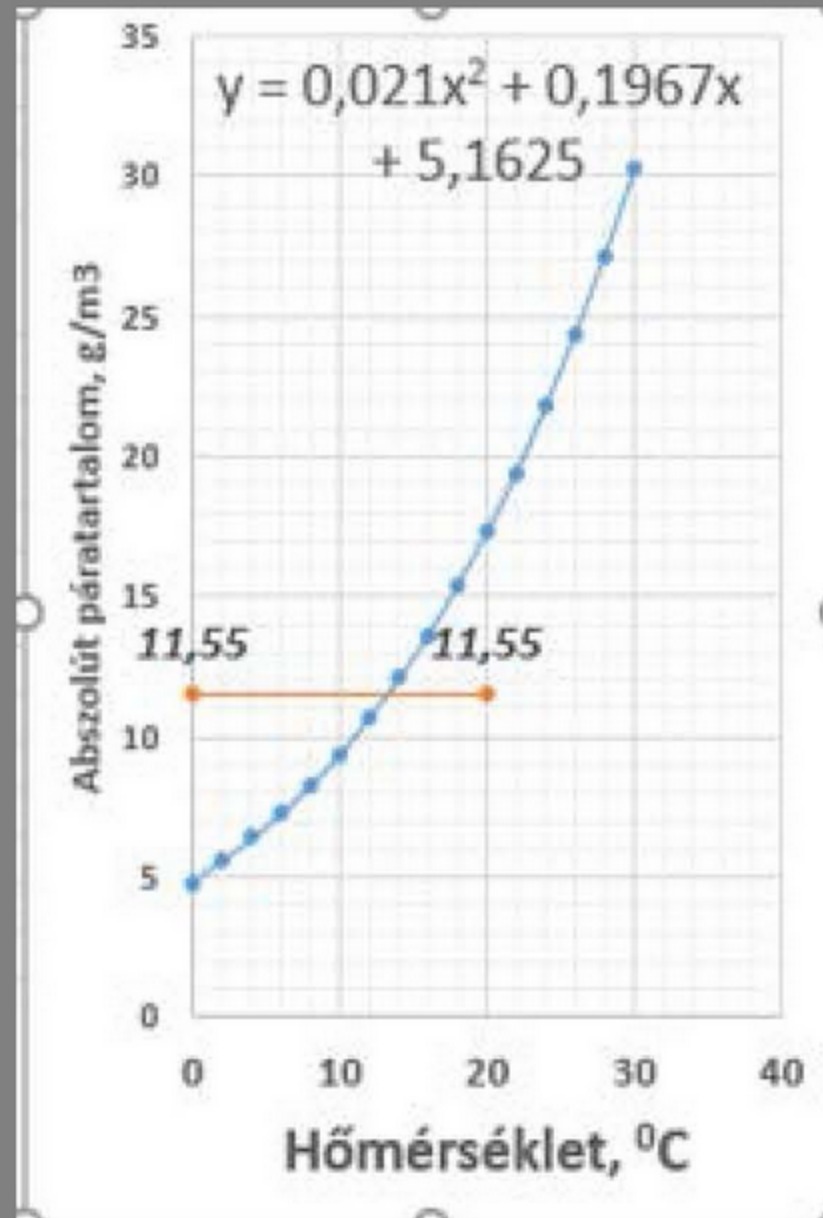

Harmatpont

Точка роси

t, Celsiusban	Abszolút páratartalom (telített) g/m ³						
0	4,8						
2	5,6						
4	6,4						
6	7,3						
8	8,3						
10	9,4						
12	10,7						
14	12,1						
16	13,6						
18	15,4						
20	17,3						
22	19,4						
24	21,8						
26	24,4						
28	27,2						
30	30,3						
Szoba hőmérséklete, Celsiusban	Abszolút páratartalom(relatív páratartalommal számítva adott hőmérsékleten) (telített) g/m ³						
20	11,55						
0	11,55						
0,00	4,56317E-05						
13,6							



$$y = 0,021x^2 + 0,1967x + 5,1625$$



$$\varphi = \frac{\rho_a}{\rho_{t.g}(t)} * 100\%$$

Abszolút páratartalom

$$\rho_a = \frac{\varphi * \rho_{t.g}(t)}{100\%}$$

Az abszolút páratartalom sűrűsége

$$0,021x^2 + 0,1967x + 5,1625 = \frac{\varphi * \rho_{t.g}(t)}{100\%}$$

$$0,021x^2 + 0,1967x + 5,1625 - \frac{\varphi * \rho_{t.g}(t)}{100\%} = 0$$

↑ ↑ ↑
a b c

$$D = b^2 - 4ac = 0,1967^2 - 4 * 0,021 * \left(5,1625 - \frac{\varphi * \rho_{t.g}(t)}{100\%}\right)$$

$$x = \frac{-0,1967 + \sqrt{(0,1967^2 - 4 * 0,021 * \left(5,1625 - \frac{\varphi * \rho_{t.g}(t)}{100\%}\right))}}{2 * 0,021}$$

Thonny - /home/pi/... 22:07

Thonny - /home/pi/Desktop/FRUZZSINA_02032020/egyben/vegso.proba.py @ 58 : 23

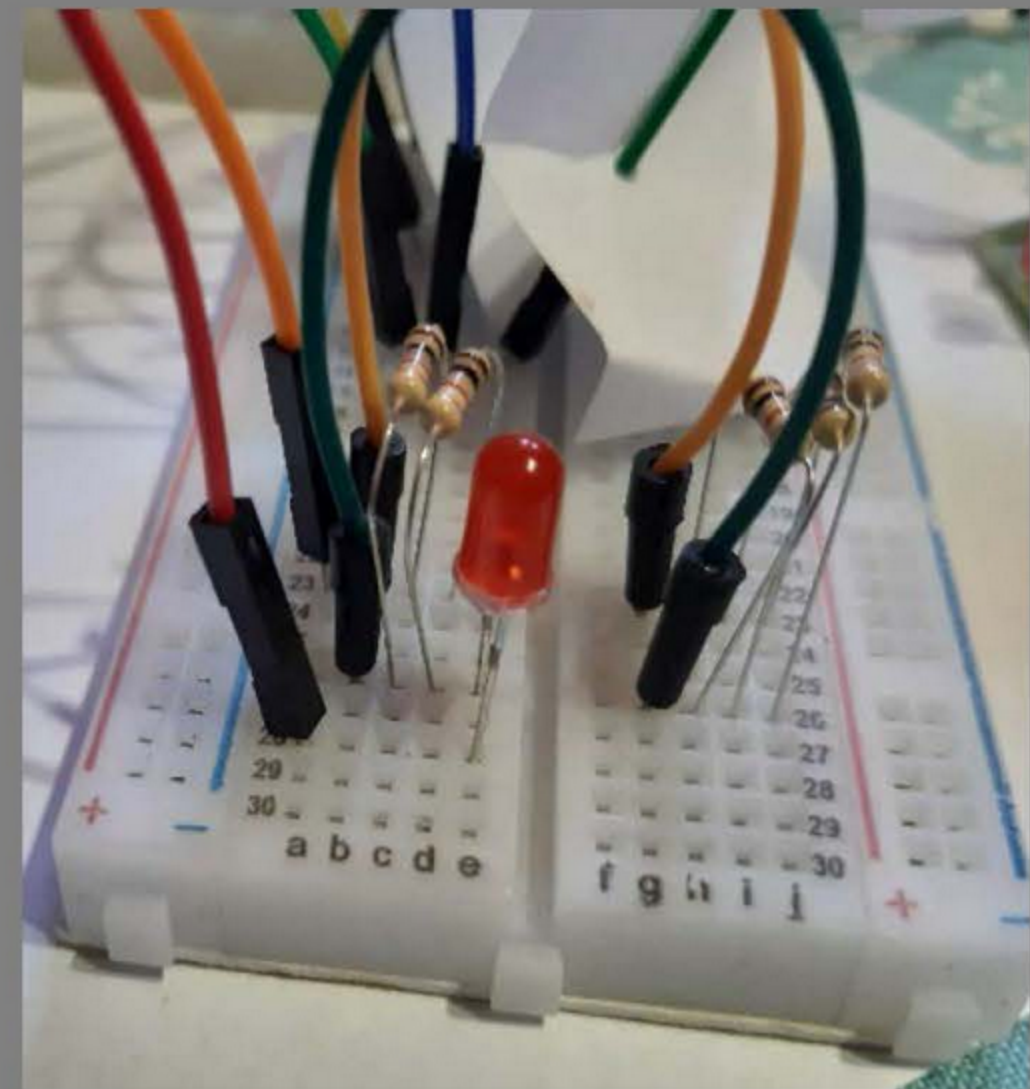
New Load Save Run Debug Over Into Out Stop Zoom Quit

vegso.proba.py 3_sensor_homero_C_K_0203.py Harmatpont_keplet_02032020.py

```
51 temp_k[i] = float(temp_c[i]) + 273
52 print ("Homero 1")
53 print(temp_c[0], "°C", temp_k[0], "K")
54 print ("Homero 2")
55 print(temp_c[1], "°C", temp_k[1], "K")
56 print ("Homero 3")
57 print(temp_c[2], "°C", temp_k[2], "K")
58 print ("Homero 4")
59 print(temp_c[3], "°C", temp_k[3], "K")
60 print("\n")
61 if temp_c[0]<x :
62     GPIO.output(18, True)
63     time.sleep(3)
64     GPIO.output(18, False)
65
66 while True:
67     h,t = dht.read_retry(dht.DHT22, DHT)
68     print('Homerseklet={0:0.2f}°C Paratartalom={1:0.2f}%'.format(t,h))
69     Stg = 0.021*t*t+0.1967*t+5.1625
70     F = 1.625 * (h*t+5) / 100
```

Shell

```
Homerseklet=20.70°C Paratartalom=48.30%
Harmatpont=9.30°C
Homero 1
19.562 °C 292.562 K
Homero 2
49.25 °C 322.25 K
Homero 3
19.25 °C 292.25 K
Homero 4
19.25 °C 292.25 K
```



Thonny - /home/pi/... 22:26

Thonny - /home/pi/Desktop/FRUZZSINA_02032020/egyben/vegso.proba.py @ 58:23

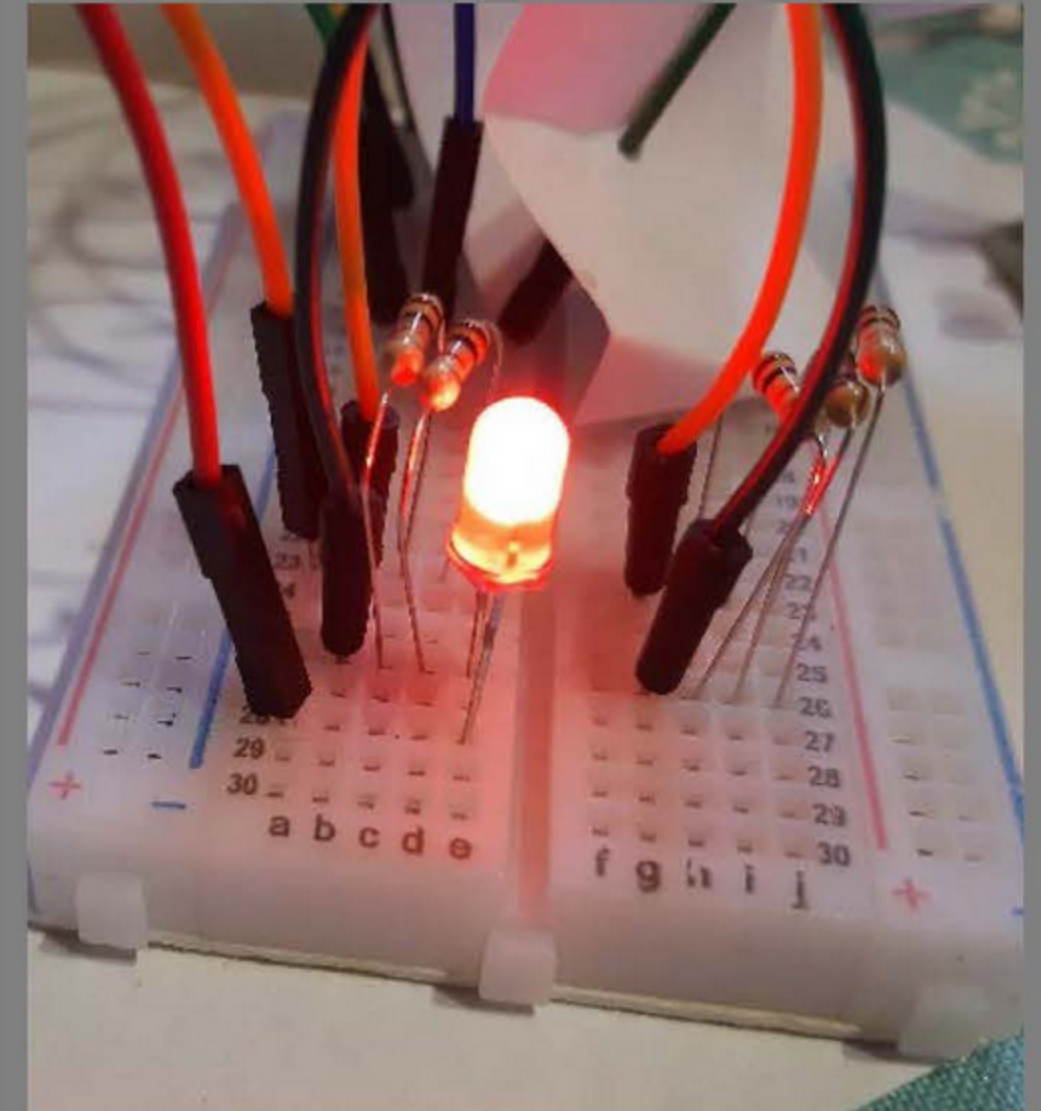
New Load Save Run Debug Over Into Out Stop Zoom Quit

vegso.proba.py

```
33 lines[3] = f3.readlines()
34 f3.close()
35 return lines
36
37 def read_temp():
38     lines = read_temp_raw()
39     while lines[0][0].strip()[-3:] != 'YES' or lines[1][0].strip()[-3:] != 'YES' or lines[
40         time.sleep(0.2)
41         lines = read_temp_raw()
42     equals_pos = [lines[0][1].find('t='), lines[1][1].find('t='), lines[2][1].find('t='), li
43     if equals_pos[0] != -1 and equals_pos[1] != -1 and equals_pos[2] != -1 and equals_pos[
44         temp_string = [lines[0][1][equals_pos[0]+2:], lines[1][1][equals_pos[1]+2:], lines[2
45         temp_c= [0,0,0,0]
46         temp_k=[0,0,0,0]
47         #return temp_c,temp_k
48
49         for i in range(0,4) :
50             temp_c[i] = float(temp_string[i]) / 1000.0
51             temp_k[i] = float(temp_c[i]) + 273
52     print ("Homero 1")
```

Shell

```
Homerseklet=20.80°C Paratartalom=50.10%
Harmatpont=9.93°C
Homero 1
9.562 °C 282.562 K
Homero 2
16.25 °C 289.25 K
Homero 3
17.187 °C 290.187 K
Homero 4
15.812 °C 288.812 K
```



MEMÓRIA-JÁTÉK KÉSZÍTÉSE DELPHI PROGRAMOZÁSI NYELVBEN

ПРОГРАМНА РЕАЛІЗАЦІЯ ІГОР ДЛЯ РОЗВИТКУ ПАМ'ЯТІ НА MOBI DELPHI

KÉSZÍTETTE: DESZKOVICS DÓRA,

AM III/6

TÉMAVEZETŐ: MAKÓ PÁL

A SZAVAK MEMORIZÁLÁSI PROGRAMJÁNAK KEZELŐFELÜLETE

УПРАВЛЯЮЧИЙ ІНТЕРФЕЙС ПРОГРАМИ ДЛЯ ЗАПАМ'ЯТОВУВАННЯ СЛІВ.

Számítógépes angol szavak memorizálása

Angol-Magyar

Networking Computers-Számítógépes hálózatok	Darab 43	Szotar
Computer-Számítógép	Darab	

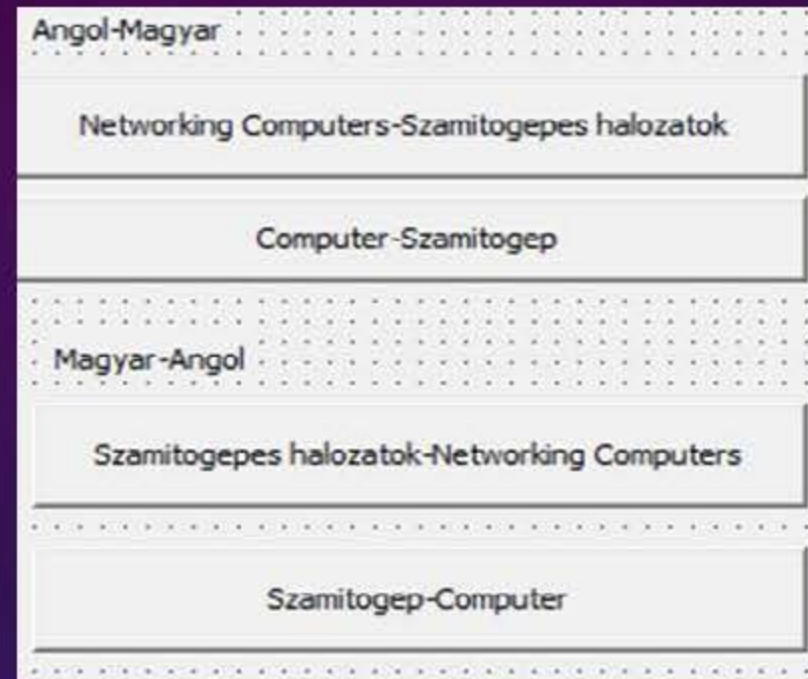
Magyar-Angol

Számítógépes hálózatok-Networking Computers	switch-kapcsoló
Számítógép-Computer	

Torles

Szavak tanulása

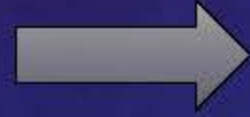
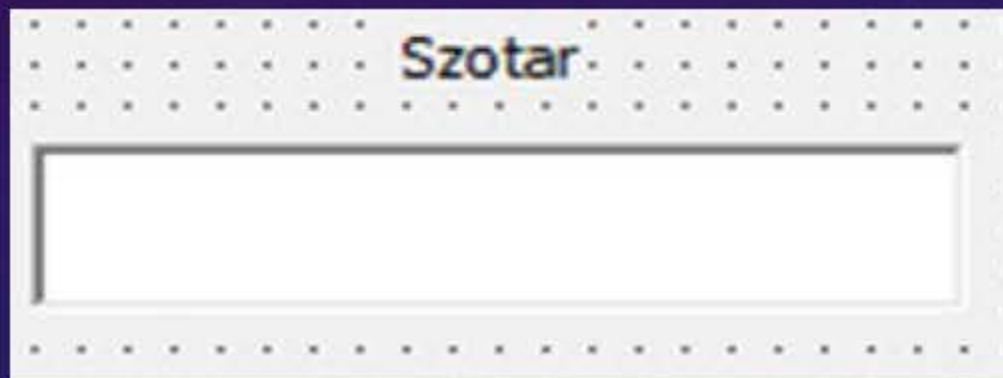
SZÁMÍTÓGÉPES ANGOL SZAVAK MEMORIZÁLÁSA ЗАПАМ'ЯТОВУВАННЯ АНГЛІЙСЬКИХ СЛІВ ЗА ДОПОМОГОЮ КОМП'ЮТЕРА



```
AssignFile (myfile, 'szavak_1_eng_hun.txt'); //  
Networking Computers (42 Words) reset  
(myfile);
```

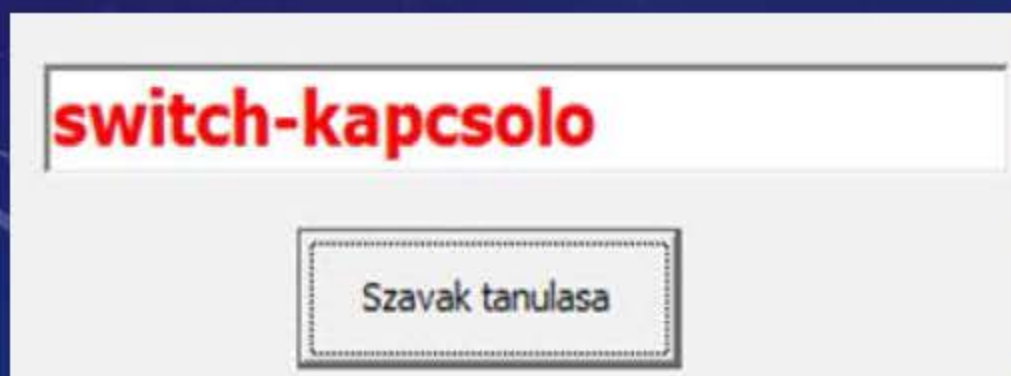
```
AssignFile (myfile, 'szavak_1_hun_eng.txt'); //  
Számítógépes hálózat (42 Words) reset (myfile);
```

Ezek a gombok arra szolgálnak, hogy beimportálják a szavakat egy txt-s fájlból



```
Edit1.Text:=Memo1.Lines[Random(i)];
```

Ebből *a többsoros szöveg megjelenítésére szolgáló komponensből*, tudja majd kiválasztani véletlenszerűen a program a szavakat



```
Edit3.Text:=inttostr(Memo1.Lines.Count);
```

```
Edit2.Text:=inttostr(Memo1.Lines.Count);
```

Szavak tanulása GOMB segítségével egyesével véletlenszerűen kiírja a szavakat fordítással

A SZÍNEK JÁTÉK PROGRAM KEZELŐFELÜLETE

УПРАВЛЯЮЧИЙ ІНТЕРФЕЙС ГРИ З КОЛЬОРАМИ

Colors-szinek

Red	Black	SkyBlue
Yellow	Blue	Fuchsia
Green	Purple	White

This is a test

Szinek változtatása Szín neve

Automatikus indítás Automatikus leállítás Kilepes

SZINEK TANULÁSA ВИВЧАННЯ КОЛЬОРІВ

Colors-szinek



begin

Panel1.color:=clRed;

→ Panel1.Caption:='Red-Piros';

end;

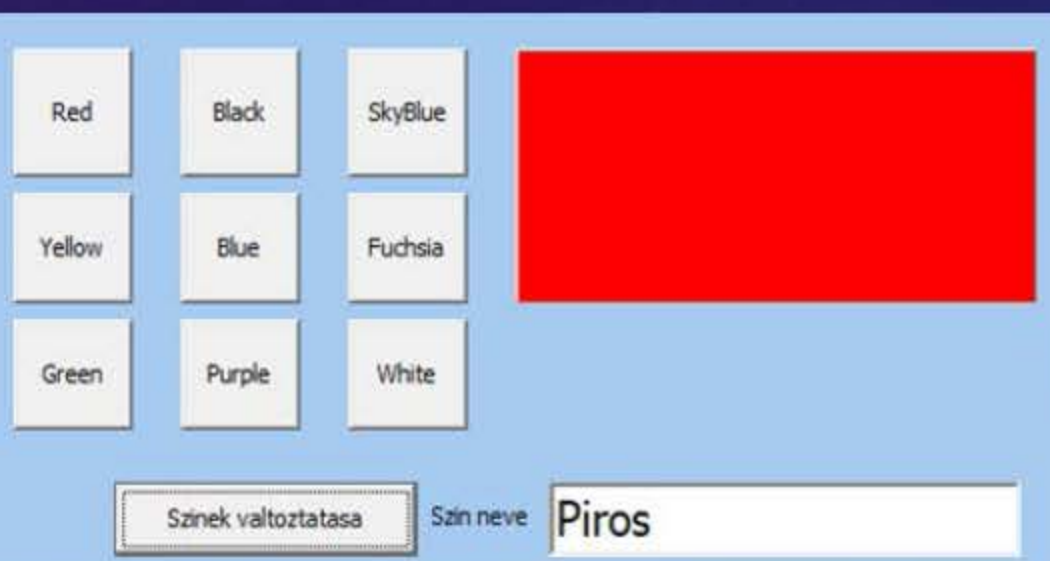
Különböző panelek, amelyekhez hozzá
kötöttem különböző színeket

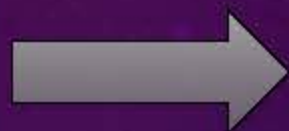


Színek megjelenítése
angol-magyar
fordítással

Panel1.Caption:=""; i:=1+Random(9);

→ Színek random megjelenítése magyar
nyelvű névvel





Időzítő

Таймер

Alkalmazásunk
periodikus
időközönként fog
valamilyen
műveletet
végezni

```
TForm1.Button12Click(Sender: TObject); begin  
Timer1.Interval:=1500;  
TForm1.FormCreate(Sender: TObject); begin  
Timer1.Interval:=0; end;
```

Automatikus indítás

Automatikus leállítás



A teszt be és -kikapcsoló gombjai

Kilépés



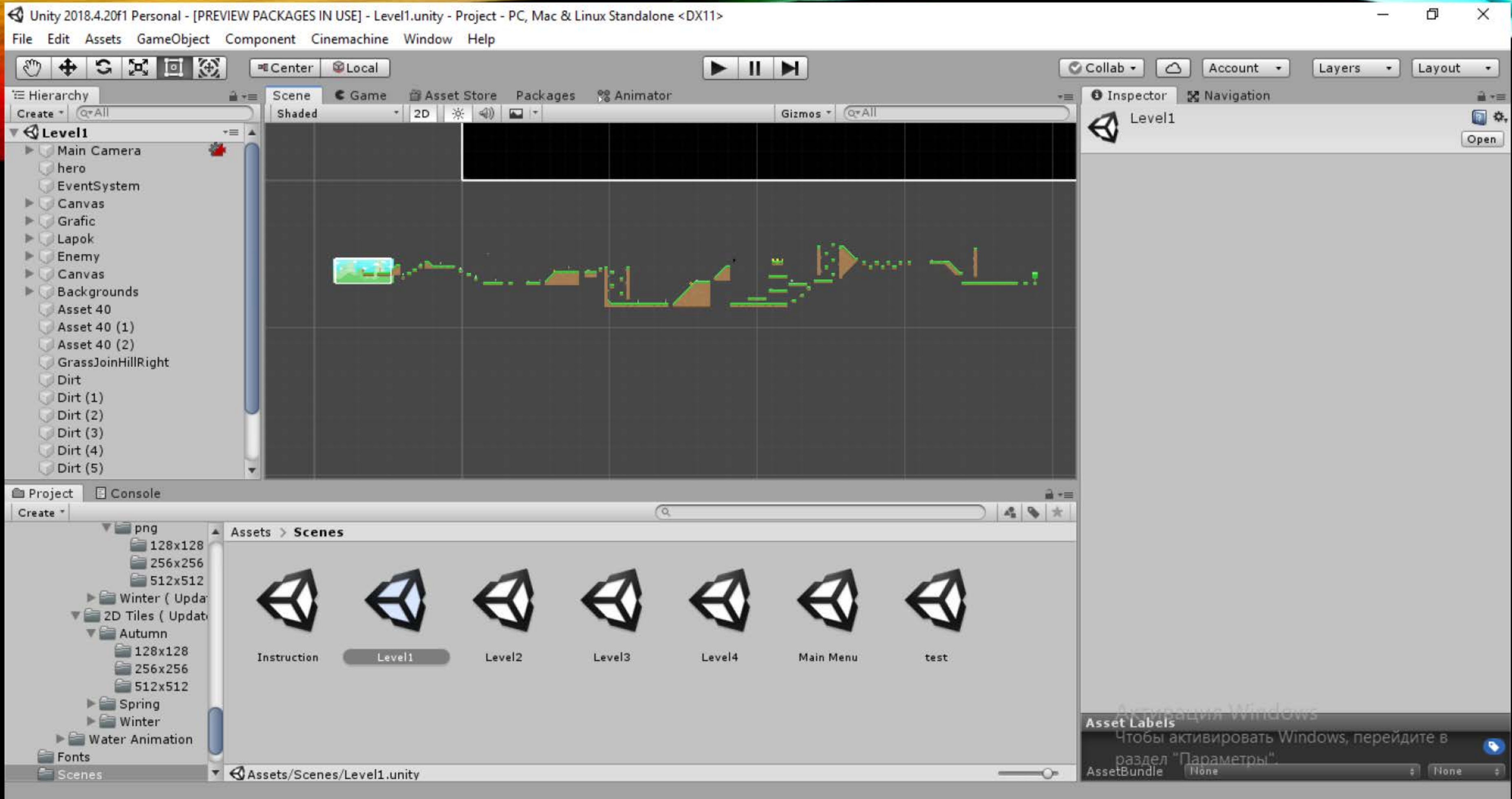
Kilépés a játékból gomb

SZÁMÍTÓGÉPES JÁTÉK TERVEZÉSE UNITY FEJLESZTŐ PLATFORMON. СТВОРЕННЯ КОМП'ЮТЕРНОЇ ГРИ НА ЦИФРОВІЙ ПЛАТФОРМІ UNITY

Készítette: Grigoryák Miklós

AM – III/6

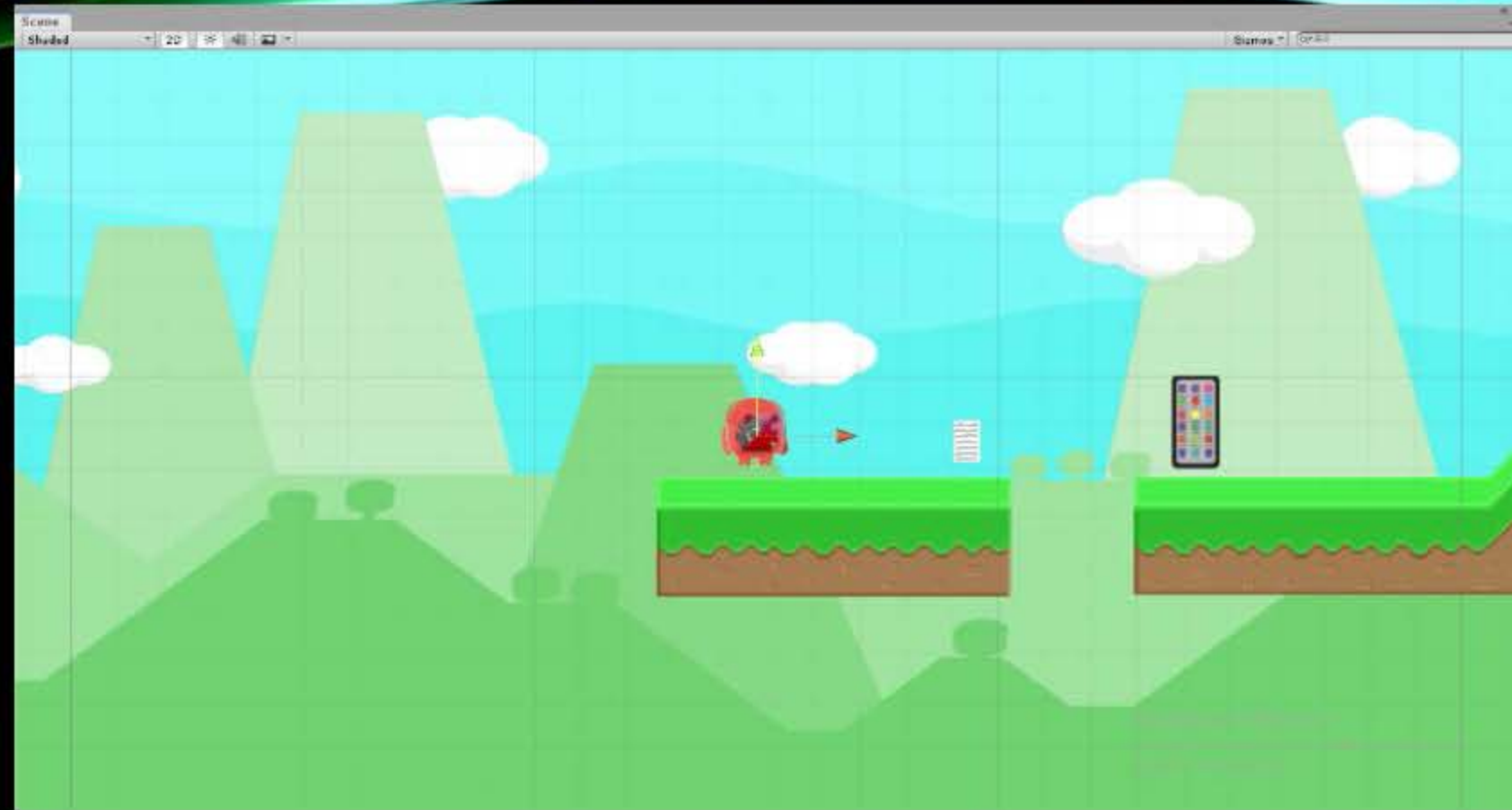
Témavezető: Szocska József



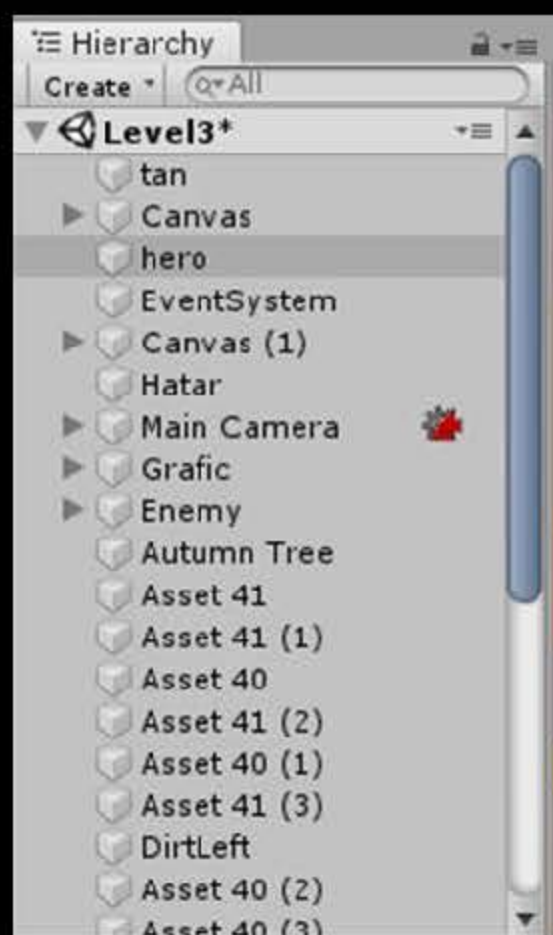
Unity interface felépítése



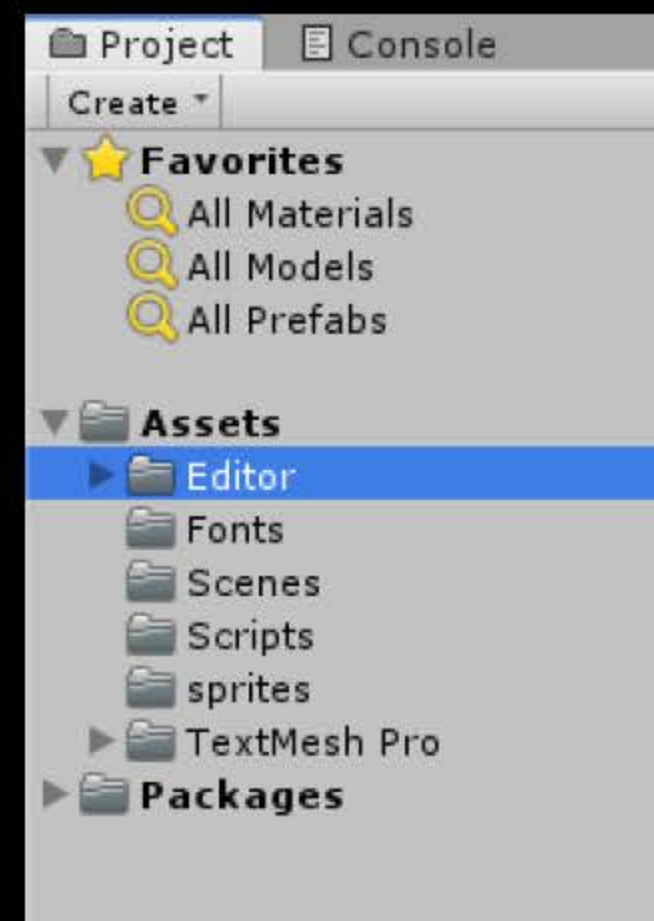
Game nézet



Scene nézet

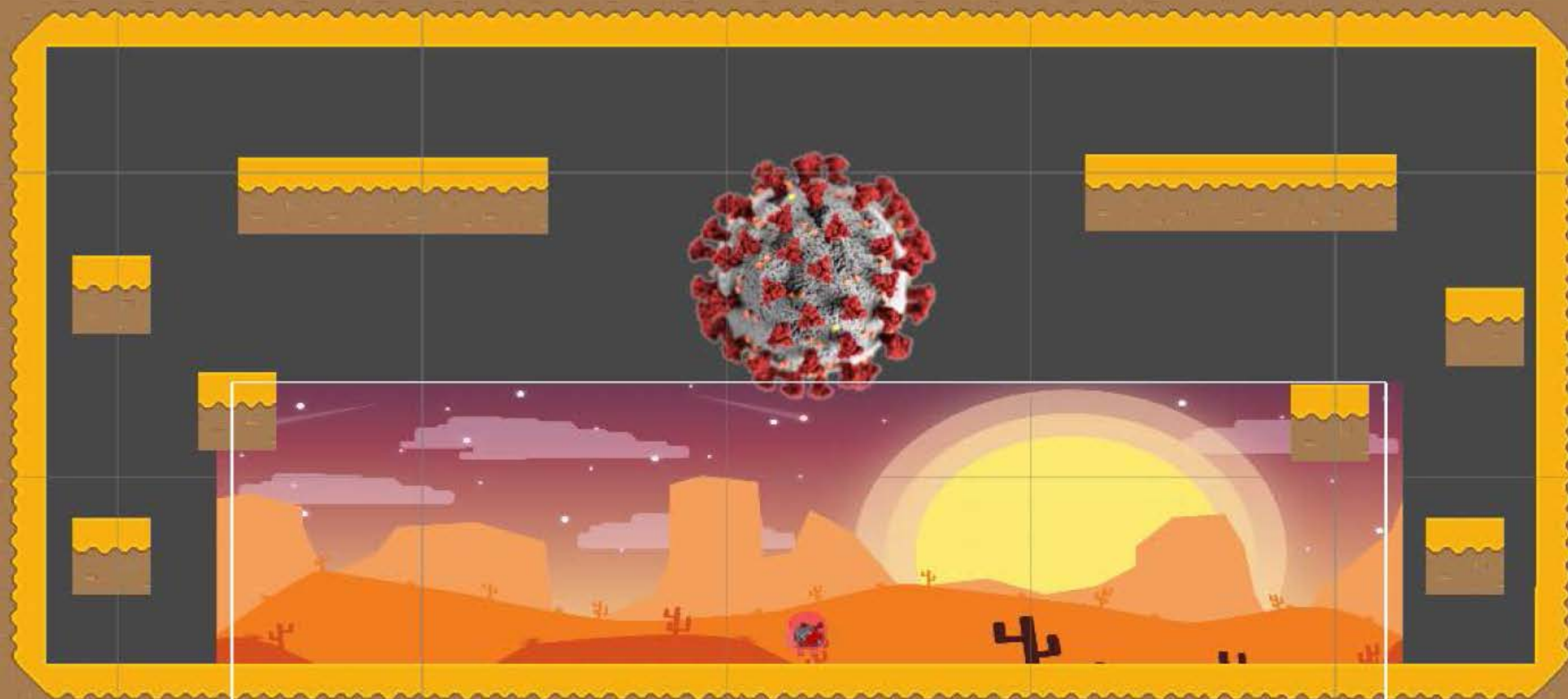


Hierarchy nézet



Project ablak

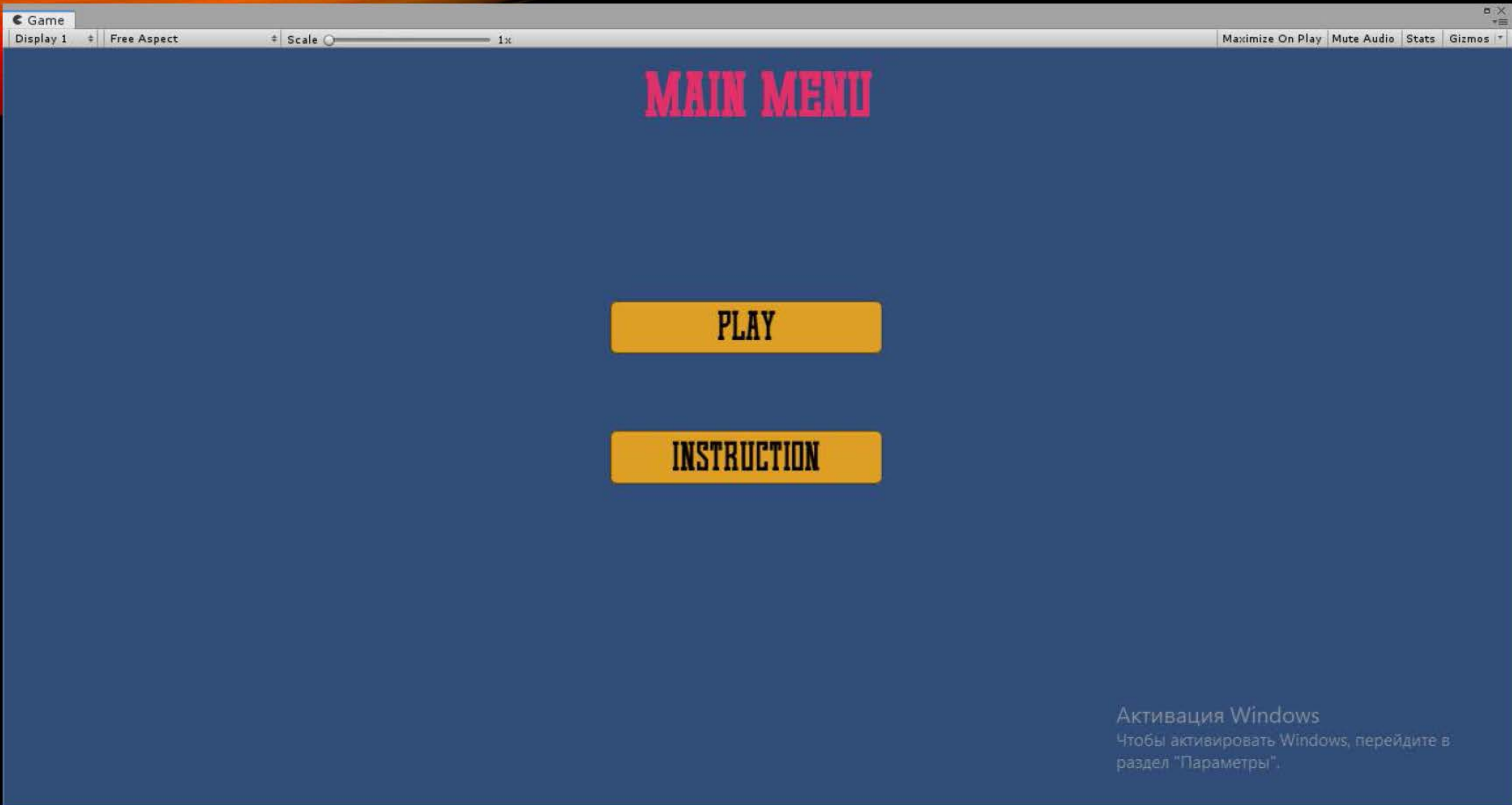




Активация Windows

Чтобы активировать Windows, перейдите в
раздел "Параметры"

Boss fight



Strukturált számítógépes hálózat kiépítése

Реалізація структурованих комп'ютерних мереж



Készítette: Buda Dániel

Szak: AM-III/6

Témavezető: Bátyi András

A szerver feladata

Завдання сервера



- Különféle szolgáltatásokat biztosít a hozzá csatlakozott számítógépnek, vagyis kliensnek.
- Optimális sebességű processzorral, felsőkategóriás RAM-mal rendelkező hűtési rendszere nagy teljesítményű, operációs rendszere fejlett és hatékony.

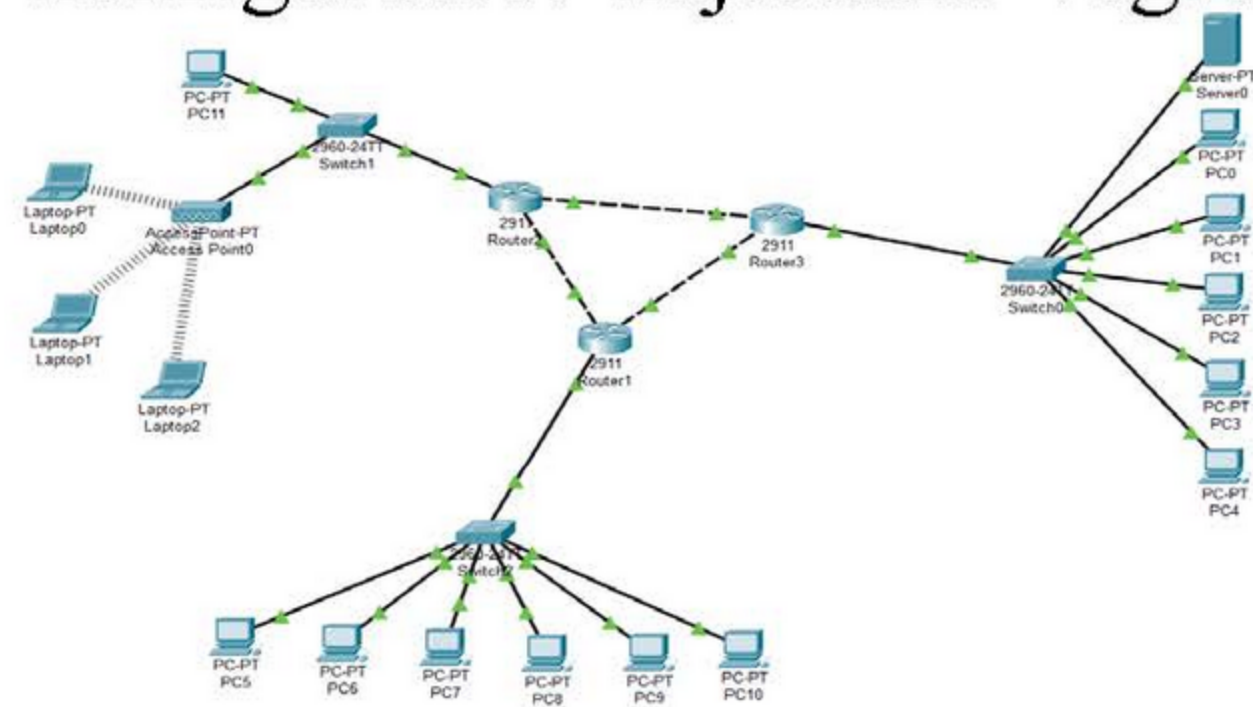


Egy lehetséges számítógép hálózat megtervezése Packet Tracerben



Проектування можливої комп'ютерної мережі в Packet Tracer

- Az volt a célom, hogy egy 3 részből álló hálózat kialakítsak, ahol minden egyes hálózati szegmens a külön alhálózathoz tartozik.
- Ennen a hálózatnak minden kliens számítógépe DHCP-n keresztül kap az adott switch-től IP beállításokat, amihez a három Routeren a DHCP konfigurációt hajtottam végre a CLI-ben.



Network

Laptop (Wi-Fi)

Virtual computer

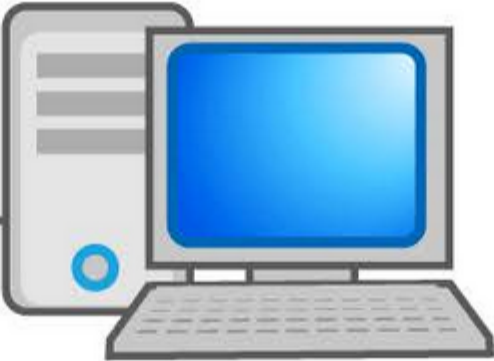


Router



Printer (Ethernet)

Smartphone (Wi-Fi)



Desktop (Ethernet)

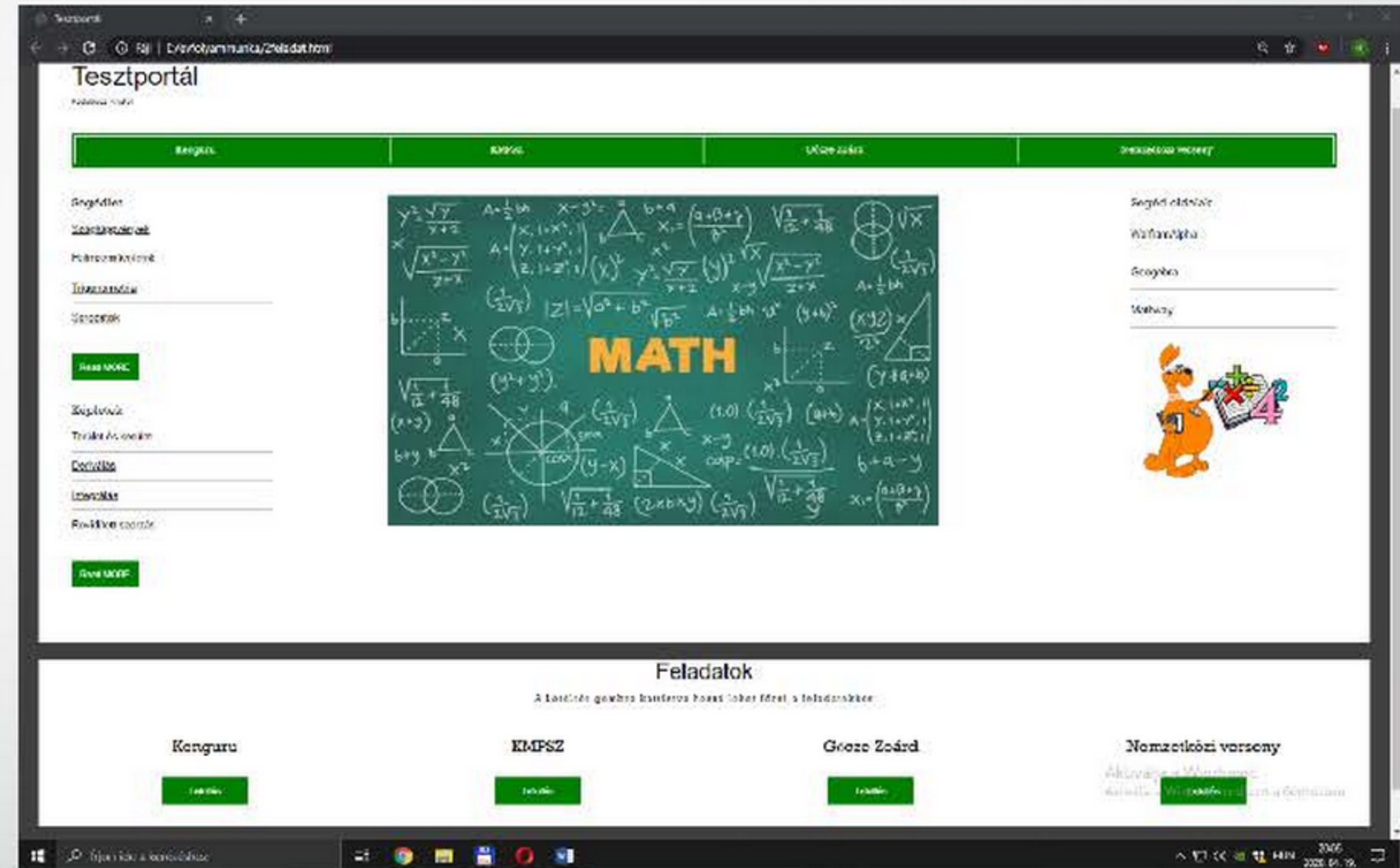
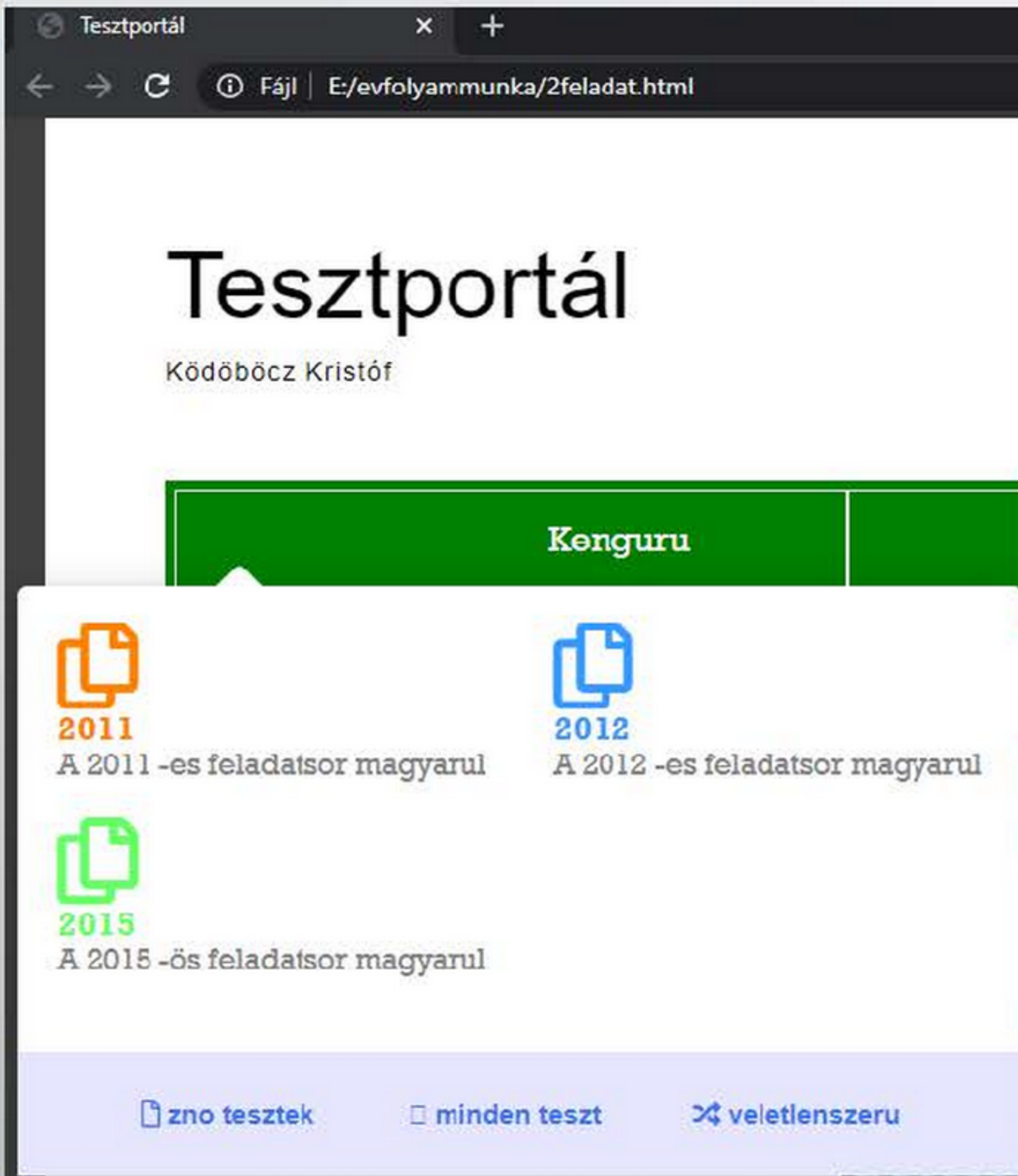


Matematika érettségi tesztportál kialakítása

Készítette: Ködöböcz Kristóf

Szak: Alkalmazott matematika III/6

Тестовий портал



GRAFIKUS FELHASZNÁLÓI FELÜLETTEL RENDELKEZŐ SZOFTVER KÉSZÍTÉSE VEKTOR MŰVELETEK VÉGZÉSÉRE

РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ З ГРАФІЧНИМ ІНТЕРФЕЙСОМ ДЛЯ РЕАЛІЗАЦІЇ ДІЙ ВЕКТОРНОЇ АЛГЕБРИ

Készítette: Orosz Diána,

AM-III/6

Témavezető: Sztojka Miroszláv, docens

A program és futása

Програма та її виконання

Skaláris szorzat kiszámítás

A kód:

```
var
  i: integer;
  s: real;
  a,b: array[1..3] of real;
begin
  for i:=1 to 3 do
    begin
      a[i]:=StrToFloat(StringGrid1.Cells[i,1]);
      b[i]:=StrToFloat(StringGrid2.Cells[i,1]);
    end;
    s:=0;
    for i:=1 to 3 do
      begin
        s:=s+a[i]*b[i];
      end;
      edit1.Text:=FloatToStr(s);
    end;
  end;
```

Bemeneti adat:

a				
	10,6	-2	25	
b				
	14	0	8	

Kimenet:

Skaláris szorzat
348,4

A program és futása

Програма та її виконання

Térfogat számítás:

Bemenet:

a				
	2	-5	3,5	

b				
	14	17,3	-4	

c				
	0	3	8	

Kimenet:

Térfogat
1007,8

A kód:

```
var
i: integer;
s: real;
a,b,c: array[1..3] of real;
begin
for i:=1 to 3 do
begin
a[i]:=StrToFloat(StringGrid1.Cells[i,1]);
b[i]:=StrToFloat(StringGrid2.Cells[i,1]);
c[i]:=StrToFloat(StringGrid3.Cells[i,1]);
end;
s:=0;
s:=abs((((a[1]* b[2]*c[3])+(a[3]* b[1]*c[2])+(a[2]* b[3]*c[1]))-
((a[3]* b[2]*c[1])+(a[1]* b[3]*c[2])+(a[2]* b[1]*c[3]))));
edit2.Text:=FloatToStr(s);
end;
```


Lineáris algebrai egyenletrendszerek iterációs numerikus programozási
megoldása módszereinek rendszerezése
Систематизація програмної реалізації чисельних ітераційних
методів розв'язування систем лінійних алгебраїчних рівнянь

Készítette: Szelley Noémi

Szak: AM III/6

Témavezető: Molnár Alexandra tanár

Jacobi iteráció megoldása C++-ban

Розв'язування ітерації Якобіана в C++

```
m=1;

line:
for(i=0;i<n;i++) {
    c[i]=b[i];
    for(j=0;j<n;j++) {
        if(i!=j) {
            c[i]=c[i]-a[i][j]*x[j];
        }
    }
}

for(i=0;i<n;i++) {
    x[i]=c[i]/a[i][i];
}

m++;

if(m<=1) {
    goto line;
}
```

A kódnak ez a része felel a Jacobi iteráció megoldásaként.

Példa

Приклад

C:\C++\jacobi\bin\Debug\jacobi.exe

Hány ismeretlenes lineáris egyenletrendszer lesz?

4

Írja be az iterációk számát:

25

Írja be a jobb oldali állandót :

3 15 27 -9

Írja le az együtthatókat :

10 -2 -1 -1

-2 10 -1 -1

1 1 10 2

-1 -1 -2 10

A megoldás :

$x(0) = 1$

$x(1) = 2$

$x(2) = 3$

$x(3) = 0$

C:\C++\jacobi\bin\Debug\jacobi.exe

Hány ismeretlenes lineáris egyenletrendszer lesz?

2

Írja be az iterációk számát:

3

Írja be a jobb oldali állandót :

3 10 -1

Írja le az együtthatókat :

2 3 -1 4

A megoldás :

$x(0) = -41$

$x(1) = -79$

Gauss-Seidel iteráció megoldása C++-ban

Ітераційне рішення Гаусса-Зайделя в C++

```
while (q> 0) {  
    for (i = 0; i < p; i++) {  
        n[i] = (b[i] / a[i][i]);  
        for (j = 0; j < p; j++) {  
            if (j == i)  
                continue;  
            n[i] = n[i] - ((a[i][j] / a[i][i]) *  
m[j]);  
            m[i] = n[i];  
        }  
    }  
}
```

A kódnak ez a része felel a Gauss Seidel iteráció megoldásaként.

Példa

Приклад

C:\C++\seidel1\bin\Debug\seidel1.exe

```
Adja meg a tomb meretet
2
a[0, 0 ]=1
a[0, 1 ]=2
a[1, 0 ]=3
a[1, 1 ]=4
Irja be a jobb oldali allandot :
b[0, 2 ]=1
b[1, 2 ]=1
Adja meg az x kezdeti ertekeit
x:[0]=0
x:[1]=0
Irja be az iteraciok szamat:
3
x1=1 x2=-0.5
x1=2 x2=-1.25
x1=3.5 x2=-2.375
```

C:\C++\seidel1\bin\Debug\seidel1.exe

```
Adja meg a tomb meretet
2
a[0, 0 ]=-1
a[0, 1 ]=-1
a[1, 0 ]=2
a[1, 1 ]=-3
Irja be a jobb oldali allandot :
b[0, 2 ]=-1
b[1, 2 ]=-1
Adja meg az x kezdeti ertekeit
x:[0]=0
x:[1]=0
Irja be az iteraciok szamat:
3
x1=-1 x2=0.333333
x1= 0.666667 x2=0.111111
x1=-0.888889 x2=0.259259
```




NEMLINEÁRIS EGYVÁLTOZÓS EGYENLETEK MEGOLDÁSA NUMERIKUS MÓDSZEREKKEL PROGRAMOZÁSI MEGOLDÁSÁNAK RENDSZEREZŐ ÁTTEKINTÉSE

KÉSZÍTETTE: TOMCSIK VANESSZA

AM III/6

TÉMAVEZETŐ: SZILÁGYI LAJOS

Szakaszfelező módszer algoritmus

Алгоритм методу розрізу навпіл

```
if(fuggveny(a) * fuggveny(b) >= 0)
{
    cout<<"a es b helytelen";
    return;
}

c=a;

while ((b-a) >= 0.01) //az intervallum elég kicsi
{
    c = (a+b)/2;
    if (fuggveny(c) == 0.0){
        break;
    }
    else if (fuggveny(c)*fuggveny(a) < 0){
        b=c;
    }
    else{
        a=c;
    }
}
```

$$X^3-2x^2+3$$

A gyök értéke = -0.998535

A húrmódszer

Рядковый метод

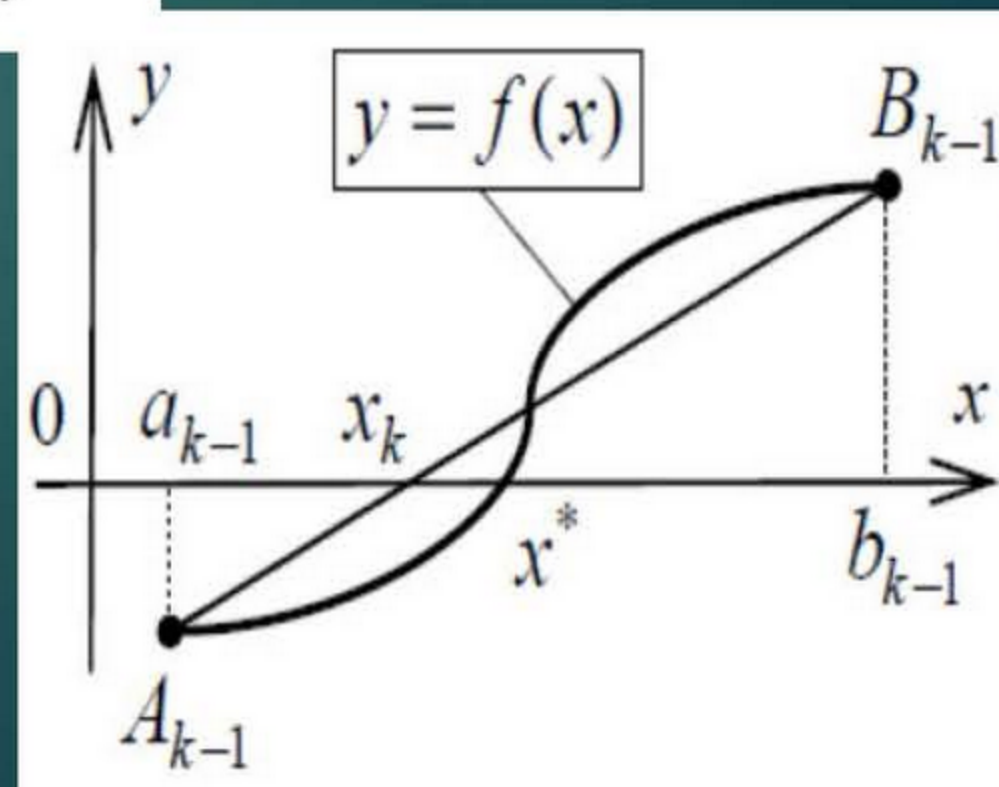
- ▶ Regula falsi = hamis módszer. (Nem a függvény, hanem a húr metszéspontját adja meg az x tengelyel.)

- ▶ A húrmódszer hatékonyabb a szakaszfelező módszerénél.

- ▶ A húr egyenlete $y - f(a) = \frac{f(b) - f(a)}{b - a} (x - a,)$

- ▶ A húrmódszer általános képlete $x_n = x_{n-1} - \frac{f(x_{n-1})(x_{n-1} - x_k)}{f(x_{n-1}) - f(x_k)}$

A húrmódszer szemléltetése



A húrmódszer algoritmus

Алгоритм рядкового методу

```
do
{
    x3=x3;
    x3=x1-(f1*(x2-x1))/(f2-f1);
    f3=fuggveny(x3);
    iter++;

    cout<<setprecision(10)<<setw(3)<<iter<<setw(15)<<x1<<setw(15)<<6,0

    if((f1*f3)<0)

        x2=x3;
    else
        x1=x3;
        f1=fuggveny(x1);
        f2=fuggveny(x2);
}while(fabs(fuggveny(x3))>=TOL); //Lezárás
```

$$x^2-4x-10$$

adja meg az x1 es x2 erteket

5		
6,0		
1	5	6 -0.2040816327
2	5.714285714	6-0.006858710562
3	5.740740741	6-0.0002289324878
4	5.741626794	6-7.639641178e-006

Lezaras :5.741656366

f(x)=-7.639641178e-006

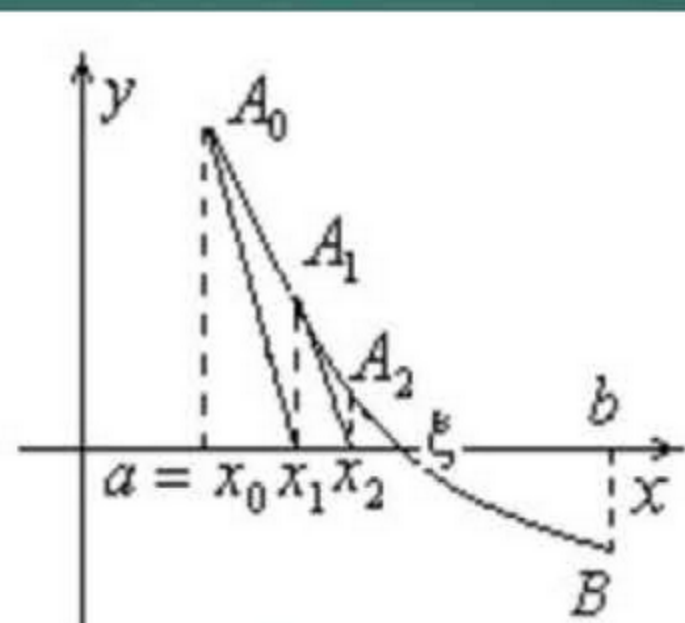
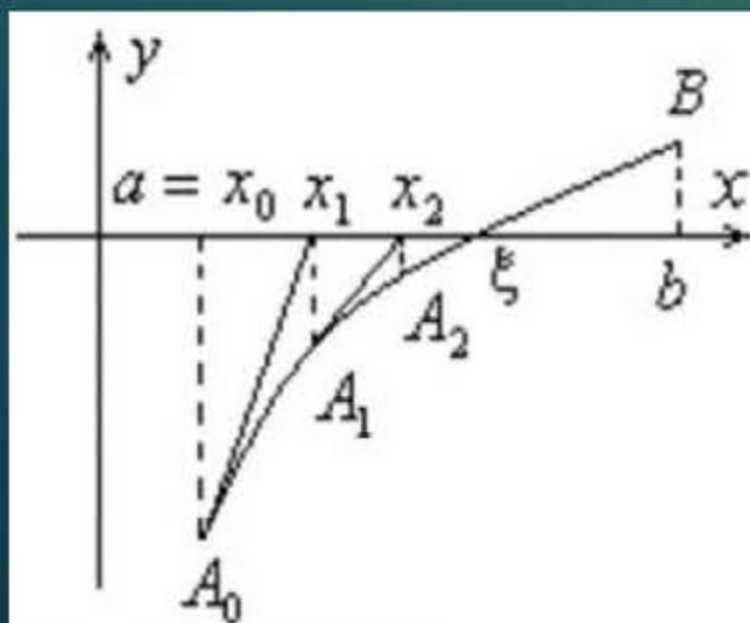
Process returned 0 (0x0) execution time : 8.270 s

Press any key to continue.

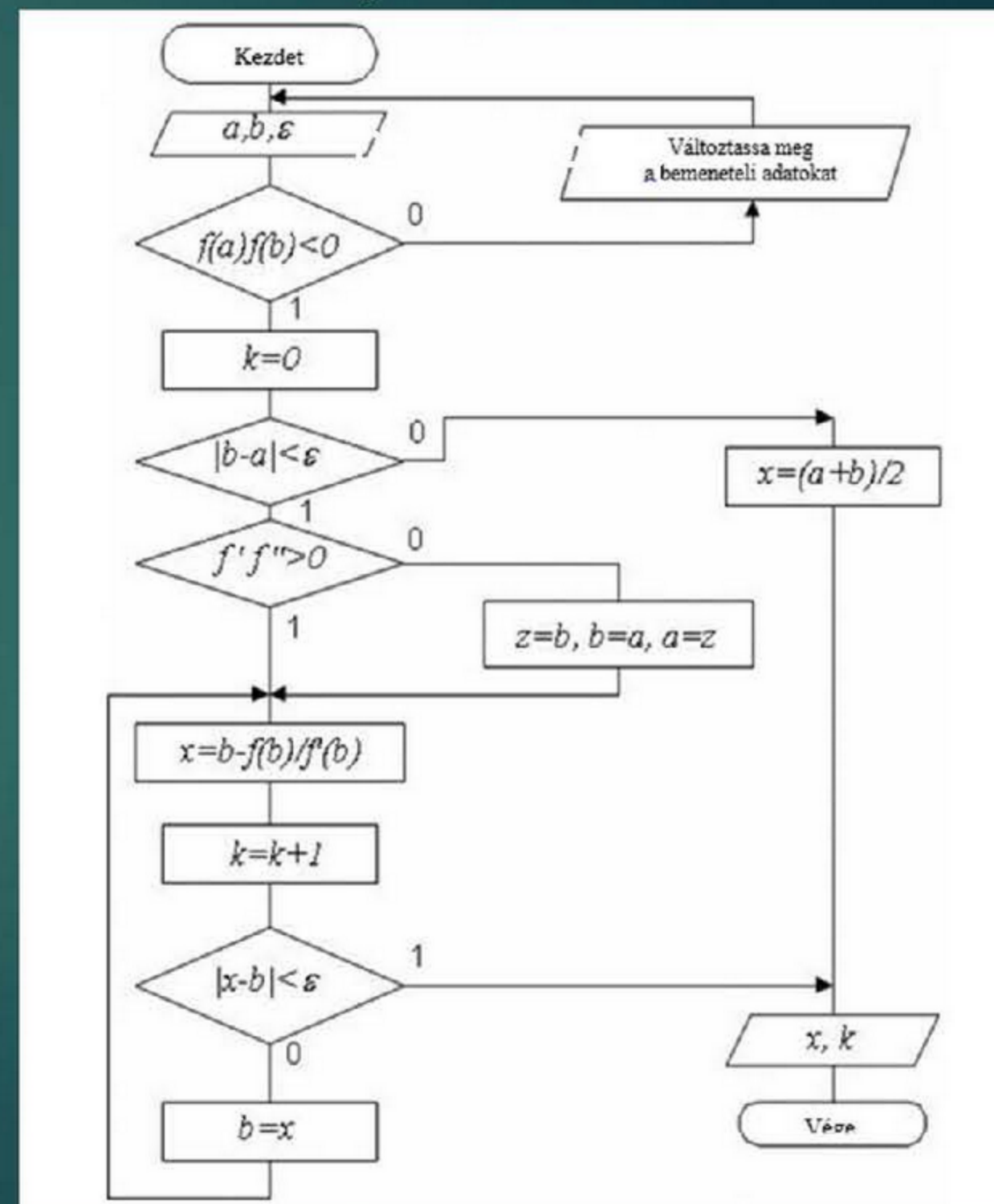
Az érintő módszer / Newton –Raphson módszer

Метод дотичної / Метод Ньютона-Рафсона

- ▶ Az egyik legjobb módszer.
- ▶ A Newton-módszer gyakran nagyon gyorsan konvergál.
- ▶ Nemcsak gyököt tudunk keresni ezen a módon, hanem minimumot vagy maximumot is találhatunk.
- ▶ A Newton-módszert először Isaac Newton írta le.



Folyamatábra az



Az érintő módszer algoritmus

Алгоритм методу дотичних

```
void newton(double x)
{
    double h = fuggveny(x) / derivFuggveny(x);
    while (abs(h) >= EPSILON)
    {
        h = fuggveny(x) / derivFuggveny(x);

        // x(i+1) = x(i) - f(x) / f'(x)
        x = x - h;
    }

    cout << "A gyok erteke : " << x;
}
```

$$x^3 - x^2 + 2$$

A gyok erteke : -1

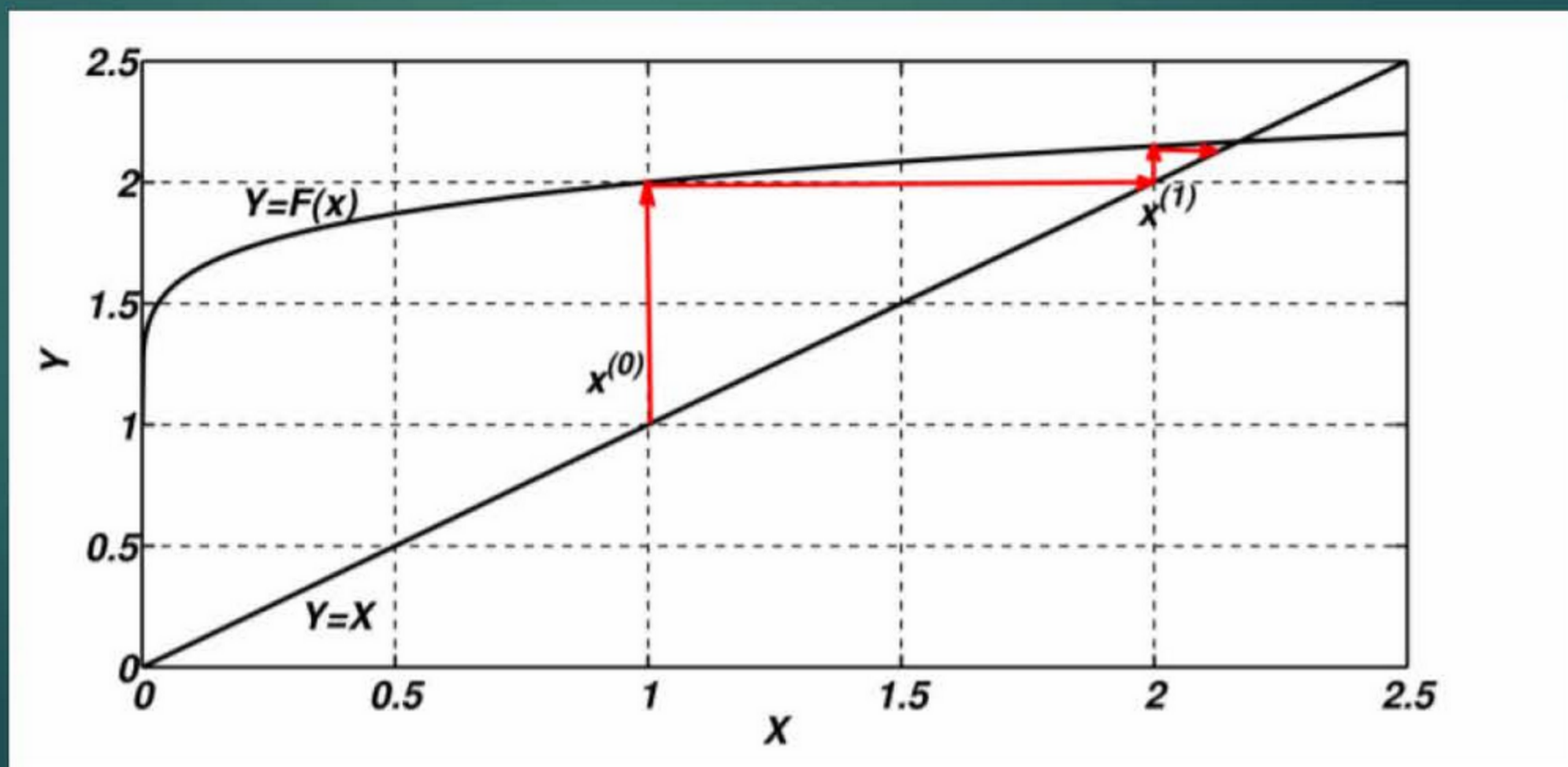
Process returned 0 (0x0) execution time : 0.056 s

Press any key to continue.

Az egyszerű iteráció módszere

Метод простой итерации

- ▶ A fixpont-iterációs egyenletmegoldás a Banach-féle fixponttételt alkalmazva határozza meg egy nemlineáris egyenlet megoldását.
- ▶ Ezt a módszert grafikusan úgy értelmezhetjük, hogy az $f(x) = 0$ egyenlet gyöke megegyezik az $y = x$ és $y = f(x) + x$ egyenletrendszer által adott vonalak metszéspontjával.



Az egyszerű iteráció módszerének algoritmus

$$\cos(x) - 3x + 1$$

Алгоритм методу простої ітерації $\cos(x) - 3x + 1$

```
do
{
    x1 = g(x0);
    cout<<"Iteracio-"<< step<<":\t x1 = "<< setw(10)<< x1<< endl;

    step = step + 1;

    if(step>N)
    {
        cout<<"Nem konvergens.";
        exit(0);
    }

    x0 = x1;

}while( fabs(f(x1)) > e);

cout<< endl<<"A gyok: "<< x1;
```

$$\cos(x) - 3x + 1$$

adja meg a kezdeti erteket: 1

adja meg a hiba erteket: 0.000001

iteraciok szama: 10

Iteracio-1: x1 = 0.513434 es f(x1) = 0.330761

Iteracio-2: x1 = 0.623688 es f(x1) = -0.059333

Iteracio-3: x1 = 0.603910 es f(x1) = 0.011391

Iteracio-4: x1 = 0.607707 es f(x1) = -0.002162

Iteracio-5: x1 = 0.606986 es f(x1) = 0.000411

Iteracio-6: x1 = 0.607124 es f(x1) = -0.000078

Iteracio-7: x1 = 0.607098 es f(x1) = 0.000015

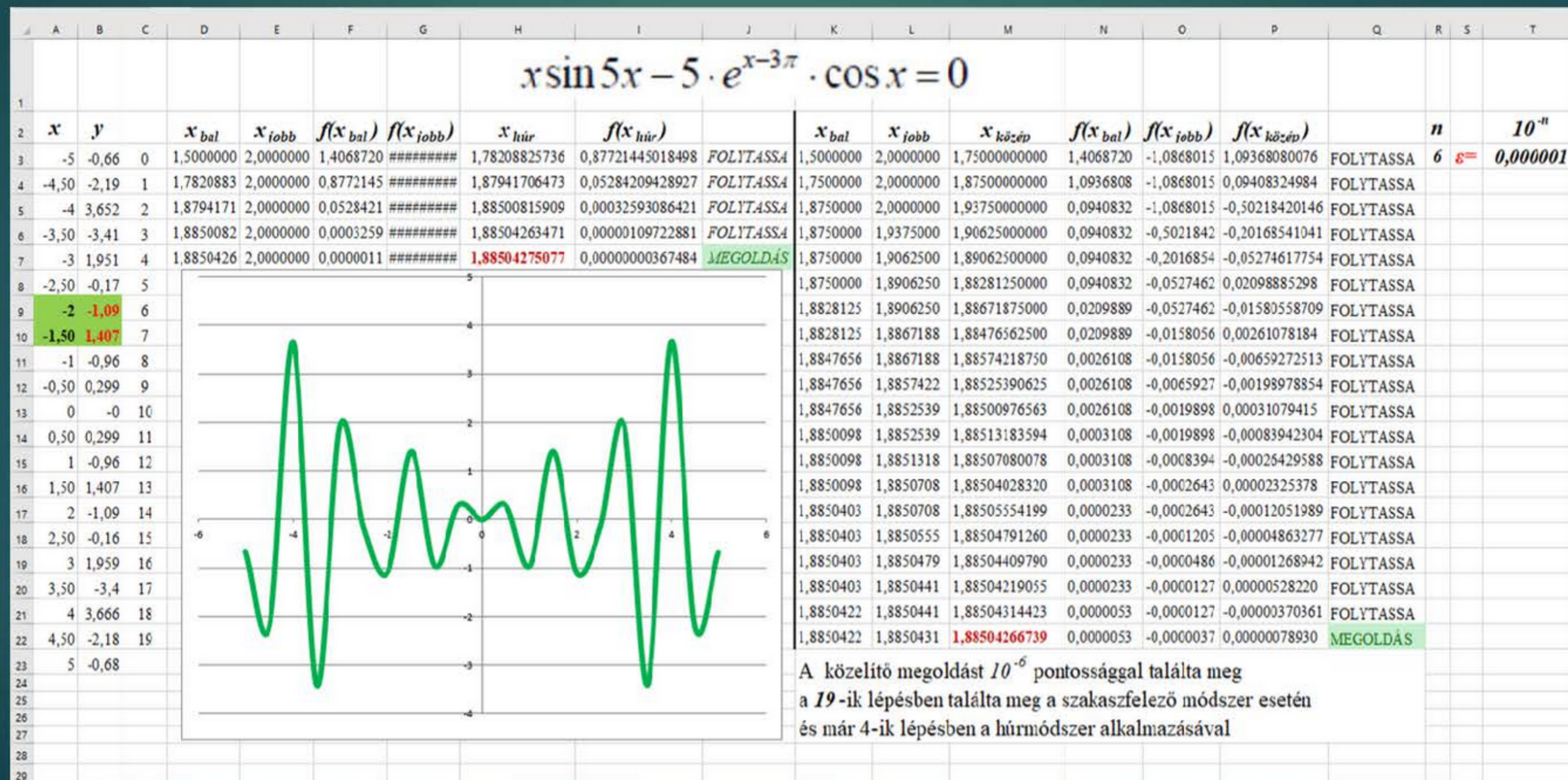
Iteracio-8: x1 = 0.607102 es f(x1) = -0.000003

Iteracio-9: x1 = 0.607102 es f(x1) = 0.000001

A gyok: 0.607102

Nemlineáris egyenletek numerikus megoldása az Excel alkalmazással

Чисельне розв'язування нелінійних рівнянь за допомогою програми Excel



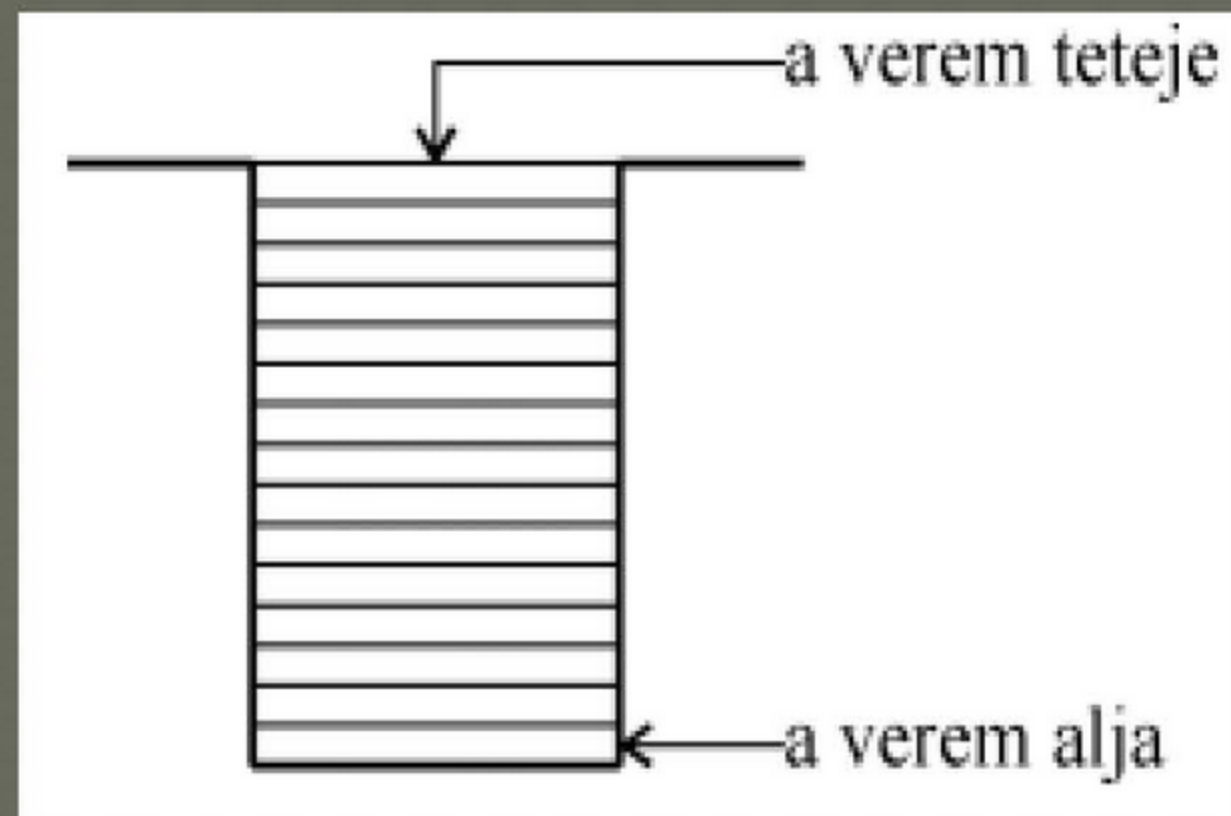
Verem, sor, lista adatszerkezetek statikus reprezentációja és implementációja

Készítette: Tóth Nikolett
AM III/6

Téma vezető: Szocska József

Verem Ctek

- A verem az egyik legegyszerűbb adatszerkezet, azonos típusú adatokat tárol.
- LIFO (Last In- First Out) adatszerkezet.



Statikus reprezentáció

Статичне представлення

```
class intverem
{
private:
    const int max=10;
    int t[10];
    int felso;
public:
    intverem();
    bool ures();
    bool tele();
    void verembe(int);
    int verembol();
};
```


Statikus implementáció

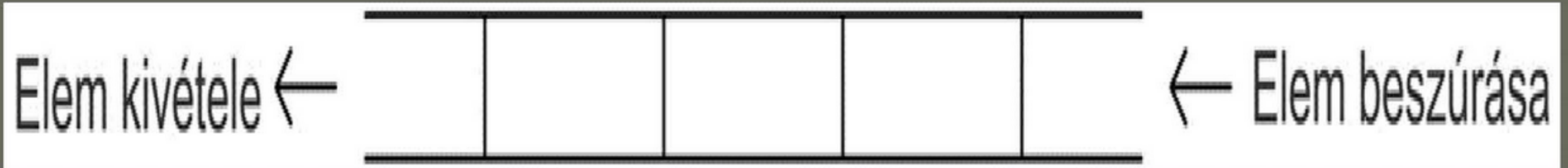
Статична реалізація

```
intverem::intverem()
{
    falso=-1;
};
bool intverem::ures()
{
    if(falso==max-1)
    {
        return true;
    }
    else
    {
        return false;
    }
};
bool intverem::tele()
{
    if(falso==max-1)
    {
        return true;
    }
    else{
        return false;
    }
};
```

```
void intverem::verembe(int ujadat)
{
    if(!tele())
    {
        falso++;
        t[falso]=ujadat;
    }
    else
    {
        cout<<"A verem tele van."<<endl;
    }
};
intverem::verembol()
{ int kivettadat;
  if(!ures())
  {
      kivettadat=t[falso];
      falso--;
  }
  return kivettadat;
};
```


Структура данных ряда

- Az elemek sorban kerülnek bele.
- FIFO (First In - First Out) szerkezet.



Statikus reprezentáció

Статичне представлення

```
class intsor
{ private:
    const int max=10;
    int t[10];
    int felso;
    int also;
    int db;
public:
    intsor();
    bool ures();
    bool tele();
    void sorba(int);
    int sorbol();
};
```


Statikus implementáció

Статична реалізація

```
intsor::intsor()
{
    db=0;
    falso=0;
    also=-1;
};
```

```
bool intsor::ures()
{
    if(db==0)
    {
        return true;
    }
    else
    {
        return false;
    }
};

bool intsor::tele()
{
    if(db==max-1)
    {
        return true;
    }
    else
    {
        return false;
    }
};
```



```

void intsor::sorba(int ujadat)
{
    if(!tele())
    {
        db++;
        if(also<max-1)
        {
            also++;
        }
        else
        {
            also=0;
        }

        t[also]=ujadat;
    }
};

```

```

int intsor::sorbol()
{ int kivettadat;
  if(!ures())
  {
      db--;
      kivettadat=t[felso];
      if(felso<max-1)
      {
          felso++;
      }
      else
      {
          felso=0;
      }
      return kivettadat;
  }
};

```


**EMAIL KÜLDÉSÉNEK AUTOMATIZÁLÁSA TÖBB
CÍMZETT SZÁMÁRA VBA PROGRAMOZÁSI
NYELVBEN**

**АВТОМАТИЗАЦІЯ НАДСИЛАННЯ
ЕЛЕКТРОННОГО ЛИСТА ДЕКІЛЬКОМ
АДРЕСАНТАМ МОВОЮ ПРОГРАМУВАННЯ VBA**

Készítette: Trombola Márk

AM III/6

Témavezető: Makó Pál

A program megvalósítása

Реалізація програми

A kellő információval a birtokomban eljutottam arra a szintre, hogy tudjak egy programot írni az Excelben létrehozott táblázat vezérléséhez VBA nyelven.

Először is létrehoztam egy Excel fájlt és a következő módon formáztam meg:

	B	C	D	E	F	G	H	I	J	K
	Vezetéknév Keresztnév	Beosztás	A személy e-mail címe akinek küldeni kell	A személy e-mail címe akinek küldeni kell (titkos másolat)	Téma	Levél alap szövege	Üzenet a levélben	Csatolt fájl_közös	Csatolt fájl_egyéni	Küldeni
1					Órarend 2019	Tisztelt VezKer!	Tisztelt !			
		kurátor	tanar0001@gmail.com			Egy órarendet és egy személyre szabott órarendet küldtünk Önnek.	Egy órarendet és egy személyre szabott órarendet küldtünk Önnek.			
2						Üdvözlettel, Trombola Márk	Üdvözlettel, Trombola Márk	E:\MARK\CSATOLT_FILE\ORAREND_EGYBEN.pdf	E:\MARK\CSATOLT_FILE\ORAREND_1_TANAR.pdf	ИСТИНА
		tanár	tanar0002@gmail.com		Órarend 2020	Tisztelt VezKer!	Tisztelt !			
						Egy órarendet és egy személyre szabott órarendet küldtünk Önnek.	Egy órarendet és egy személyre szabott órarendet küldtünk Önnek.			
3						Üdvözlettel, Trombola Márk	Üdvözlettel, Trombola Márk	E:\MARK\CSATOLT_FILE\ORAREND_EGYBEN.pdf	E:\MARK\CSATOLT_FILE\ORAREND_2_TANAR.pdf	ИСТИНА
					Órarend 2021	Tisztelt VezKer!				

Egy kis táblázatot hoztam létre , amelybe beszúrtam két makró gombot. **Az egyik az e-mailek küldésére**, a másik pedig az e-mailek **elküldésének állapotát jelző oszlop törlésére** szolgál. Ezen kívül ebben a táblázatban kiválaszthatjuk , hogy kinek szeretnénk elküldeni a levelet. Egy képlet segítségével lehetővé tettem , hogy külön lehessen küldeni a levelet minden csoportnak .

E-mail küldése
mindenkinek

TÖRLÉS

☒ Tanár

☒ Kurátor

☒ Szakbizottság

=ИЛИ(ЕСЛИ(C2=\$R\$2;1;0);ЕСЛИ(C2=\$R\$3;1;0);ЕСЛИ(C2=\$R\$4;1;0))

I	J	K
Csatolt fájl_közös	Csatolt fájl_egyéni	Küldeni
MARK\CSATOLT_FILE\ORAREND_EGYBEN.pdf	E:\MARK\CSATOLT_FILE\ORAREND_1_TANAR.pdf	ИСТИНА

Utolsó lépésként pedig a két kód megírása volt VBA programozási nyelven. Miután a program lefűt „Mindenkinek sikeresen elkűldve” űzenet jelenik meg a k  perny  n.

K4 X ✓ fx =ИЛИ(ЕСЛИ(C4=\$R\$2;1;0);ЕСЛИ(C4=\$R\$3;1;0);ЕСЛИ(C4=\$R\$4;1;0))

	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
	Lev��l alap sz��vege	��zenet a lev��lben	Csatolt f��jl_k��z��s	Csatolt f��jl_egy��ni	K��ldeni	��llapot						Seg��d t��bl��zat		
1	Tisztelt VezKer!	Tisztelt !												
	Egy ��rendet ��s egy szem��lyre szabott ��rendet k��ldt��nk ��nnek.	Egy ��rendet ��s egy szem��lyre szabott ��rendet k��ldt��nk ��nnek.												
2	��dv��zlettel, Trombola M��rk	��dv��zlettel, Trombola M��rk	E:\MARK\CSATOLT_FILE\ORAREND_EGYBEN.pdf	E:\MARK\CSATOLT_FILE\ORAREND_1_TANAR.pdf	ИСТИНА	K��ldve								
	Tisztelt VezKer!	Tisztelt !												
	Egy ��rendet ��s egy szem��lyre szabott ��rendet k��ldt��nk ��nnek.	Egy ��rendet ��s egy szem��lyre szabott ��rendet k��ldt��nk ��nnek.												
3	��dv��zlettel, Trombola M��rk	��dv��zlettel, Trombola M��rk	E:\MARK\CSATOLT_FILE\ORAREND_EGYBEN.pdf	E:\MARK\	ИСТИНА	K��ldve								
	Tisztelt VezKer!	Tisztelt !												
	Egy ��rendet ��s egy szem��lyre szabott ��rendet k��ldt��nk ��nnek.	Egy ��rendet ��s egy szem��lyre szabott ��rendet k��ldt��nk ��nnek.												
4	��dv��zlettel, Trombola M��rk	��dv��zlettel, Trombola M��rk	E:\MARK\CSATOLT_FILE\ORAREND_EGYBEN.pdf	E:\MARK\	ИСТИНА	K��ldve								
	Tisztelt VezKer!	Tisztelt !												
	Egy ��rendet ��s egy szem��lyre szabott ��rendet k��ldt��nk ��nnek.	Egy ��rendet ��s egy szem��lyre szabott ��rendet k��ldt��nk ��nnek.												

E-mail k  ld  se mindenk  nek

T  RL  S

☒ Tan  r

☒ Kur  tor

☒ Szakbizotts  g

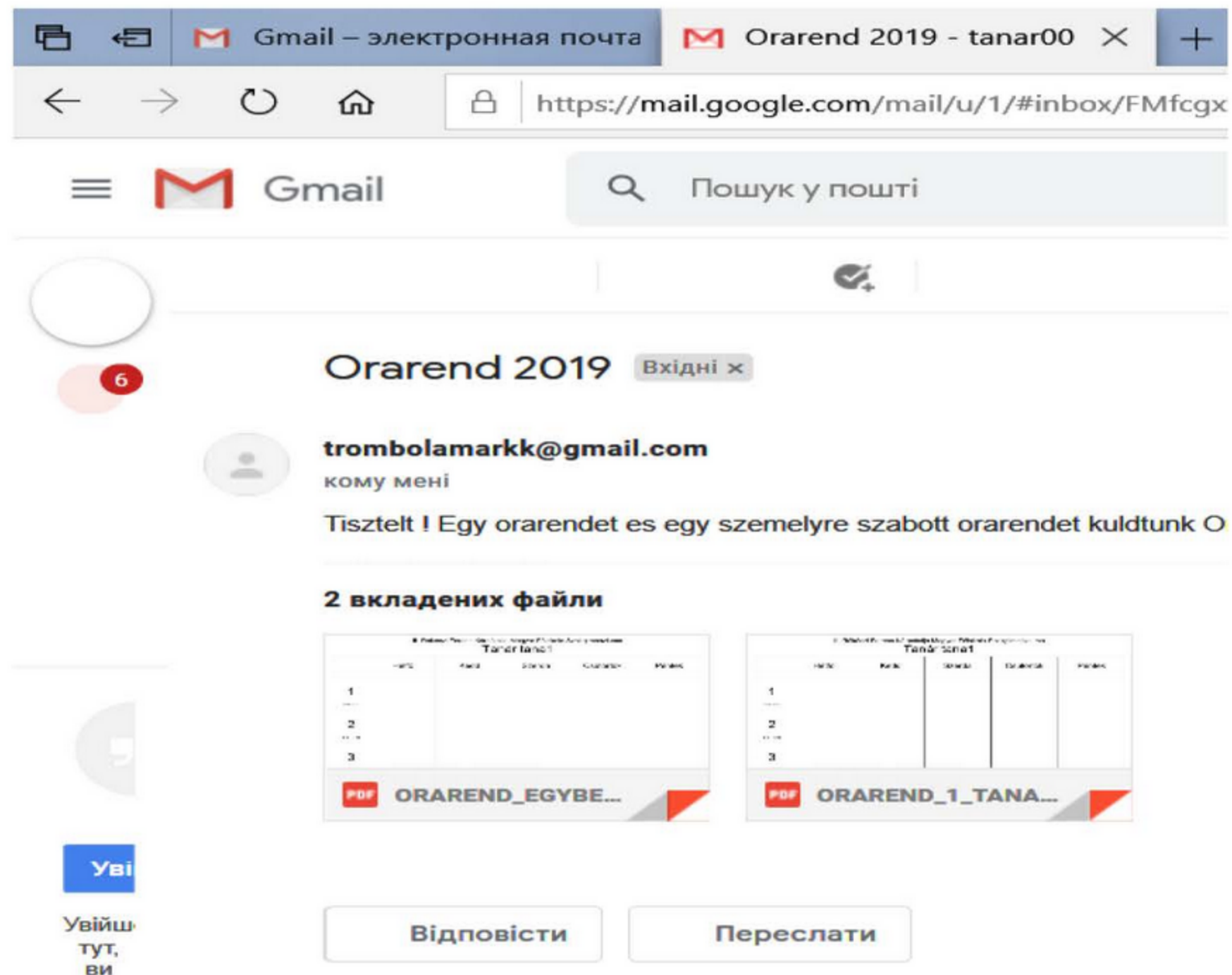
Microsoft Excel

Mindenk  nek sikeresen elk  ldve

OK

kur��tor	ИСТИНА	kur��tor
szakbizotts��g	ИСТИНА	szakbizotts��g
tan��r	ИСТИНА	tan��r

Végeredmény: Кінцевий результат:



II. Rákóczi Ferenc Kárpátaljai Magyar Főiskola Szakgimnáziuma

**SZÁMÍTÓGÉP ÁLTAL VEZÉRELT TÖBB KOMPONENSŰ
BIZTONSÁGTECHNIKAI RENDSZER KIALAKÍTÁSA**
РОЗРОБКА ЦИФРОВОГО УПРАВЛІННЯ БАГАТОКОМПОНЕНТНОЇ СИСТЕМИ БЕЗПЕКИ

Készítette: Tóth Vivien

III. évfolyamos Alkalmazott matematika

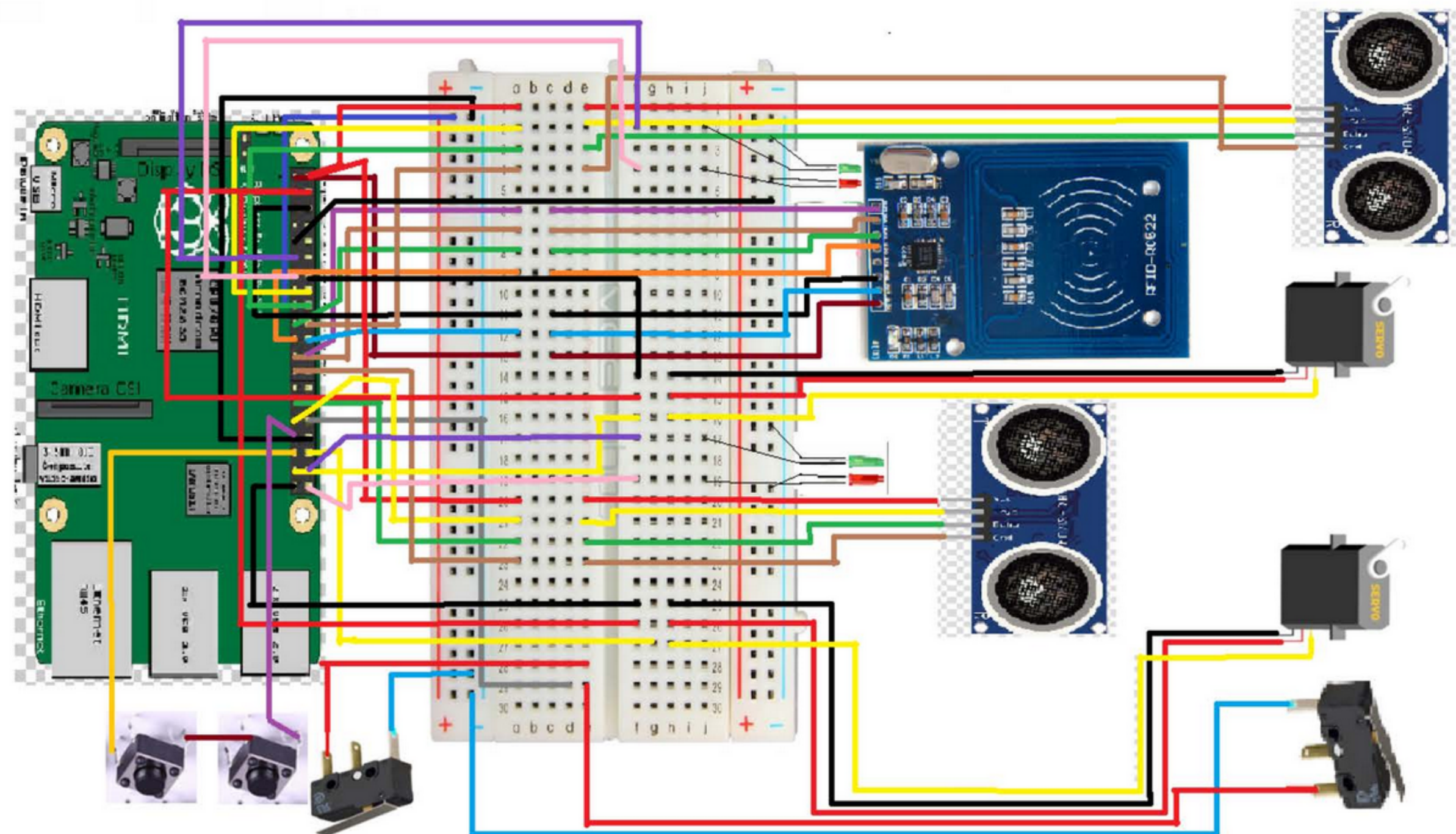
szakos hallgató

Témavezető: Simon Lénárd

tanár

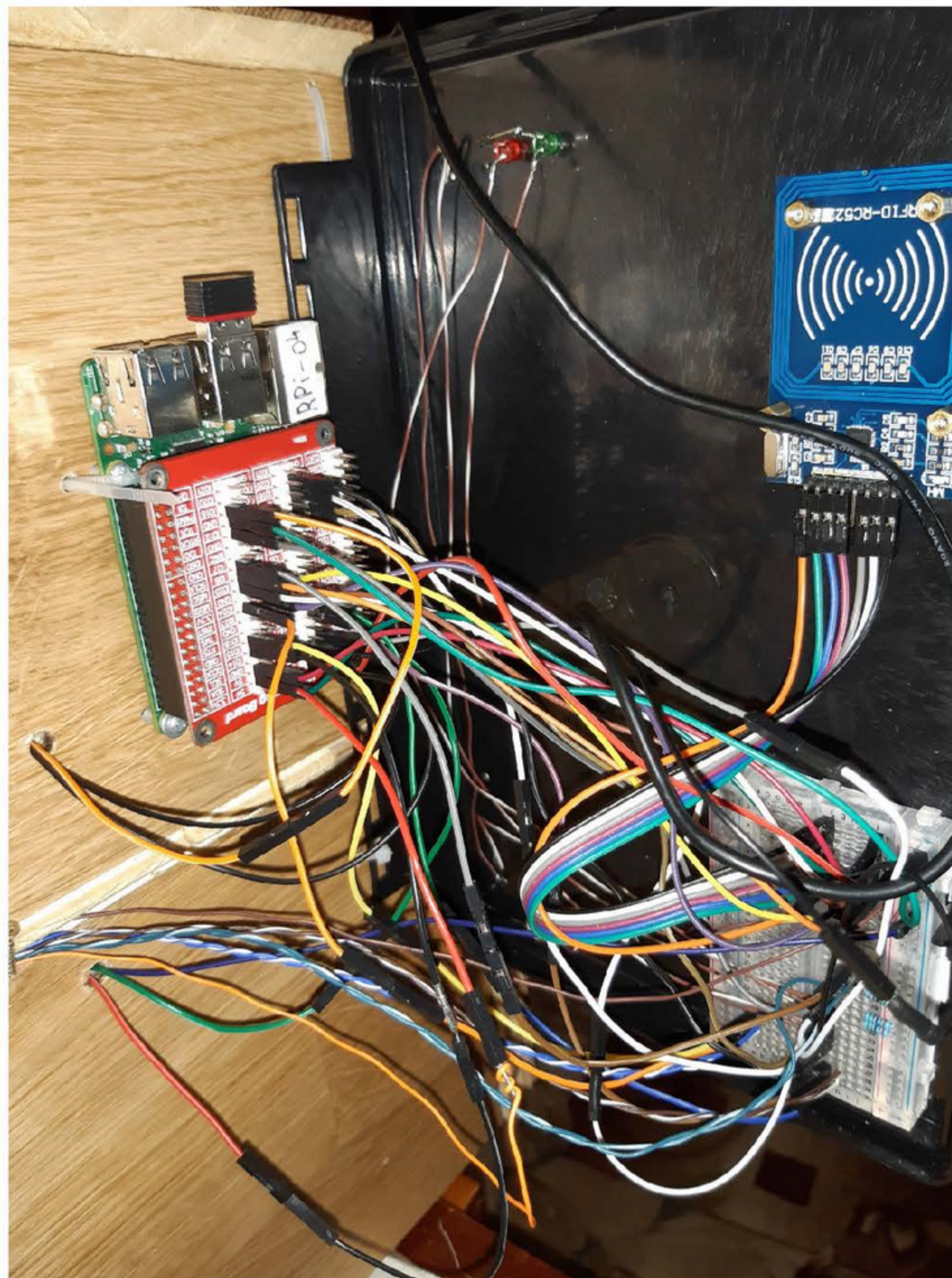
Összekötési rajz

Схема підключення



**Gyakorlatban a rendszer
összekötése**

**Підключення системи на
практиці**





Robotkar programozása számok írására Delphiben Побудова та програмування в операційній системі Windows системи робота-машини для запису цифр

Készítette: Sárközi Erik

AM III/6

Témavezető: Makó Pál

Az igazság táblázata mind két motornak

Таблиця істинності для обох двигунів

Igazságtáblázat			
1	0	0	1
0	0	1	1
0	1	1	0
1	1	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
9	12	6	3

Az első motor
igazságtáblázata

Igazságtáblázat			
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
1	0	0	1
0	0	1	1
0	1	1	0
1	1	0	0
144	192	96	48

A második motor
igazságtáblázata

A program kezelőfelülete:

Інтерфейс користувача програми:

The application window, titled 'Form1', features a blue background and a 'KILEPES' button in the top-left corner. The main title 'Számok írása 0-9-ig' is centered at the top. Below this, the interface is divided into two motor control sections and a numeric keypad.

1. Motor

- ☐ Kék
- ☐ Rózsaszín
- ☐ Sárga
- ☐ Narancs

2. Motor

- ☐ Kék
- ☐ Rózsaszín
- ☐ Sárga
- ☐ Narancs

Navigation buttons for the motors are located below the color lists:

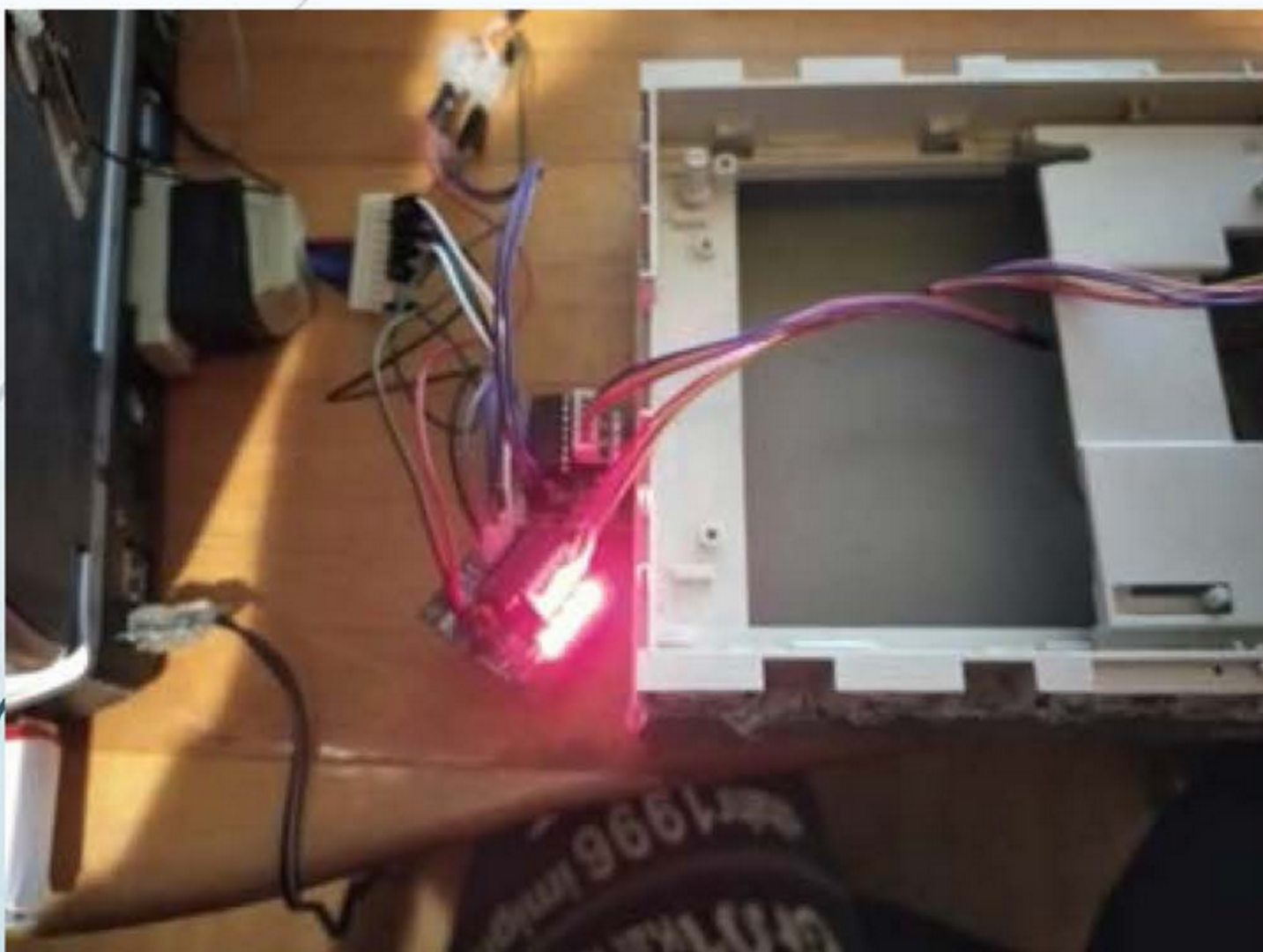
- Motor 1: 'előre' and 'hátra' buttons.
- Motor 2: 'Hatra' and 'előre' buttons.

A numeric keypad is positioned to the right of the motor controls, containing buttons for digits 0 through 9.

At the bottom, a label 'teljes forgas db' is followed by a text box containing the value '1' and a horizontal slider control.

A vezérlő a program futása közben

Керування під час виконання програми



A robotkar a program futása közben

Роботична рука під час виконання програми

