

Alkalmazások fejlesztése, projekt labor I. (Lev)

Féléves beadandó feladat

Készítette:

Pisztró Dániel

(ELJUVY)

Sátoraljaújhely, 2025.05.13.

Leírás

Az elkészült program egy Neptunhoz hasonló rendszert hivatott bemutatni. Jelen verziójában a felhasználó képes bejelentkezni **adminisztrátorként** vagy **tanulóként**.

Az adminisztrátorok képesek bármely tanuló státuszát megváltoztatni, felvenni vagy leadni bármely tanulót bármely kurzusról, kilistázni a tanulókat és adataikat, valamint a kurzusokat.

A tanulók képesek saját státuszukat megváltoztatni, felvenni, valamint leadni kurzusokat és kilistázni minden elérhető kurzust.

A forráskód részletei

Létrehozott osztályok:

Person: Szülő osztály, amely tartalmazza a közös adattagokat (Név, ID)

```
3 references
abstract class Person
{
    9 references
    public String Name { get; set; }
    14 references
    public string Id { get; set; }
}
```

Student: Tanulók osztálya, amely tartalmazza a Nevüket, Státuszukat, ID-t valamint Kreditjüket

```
class Student : Person
{
    6 references
    public Boolean Status { get; set; }
    1 reference
    public int Credits { get; set; }
    3 references
    public Student(Boolean status, String name, string id, int credits)
    {
        this.Status = status;
        this.Name = name;
        this.Id = id;
        this.Credits = credits;
    }
}
```

Teacher: Tanárok osztálya, amely a tanár nevét és ID-jét tartalmazza

```
8 references
class Teacher : Person
{
    2 references
    public Teacher(String name, string id)
    {
        this.Name = name;
        this.Id = id;
    }
}
```

Admin: Adminisztrátorok osztálya, amely az adminisztrátor nevét és ID-jét tartalmazza

```
class Admin : Person
{
    3 references
    public Admin(String name, string id)
    {
        this.Name = name;
        this.Id = id;
    }
}
```

Course: Kurszok osztálya, amely tartalmazza a kurzust tartó Tanárt (mint objektumot), a kurzus nevét, a kurzus ID-jét, kredit értékét, valamint a jelentkezett tanulók ID-jét

```
14 references
class Course
{
    1 reference
    public Teacher Teacher { get; set; }
    5 references
    public String Name { get; set; }
    4 references
    public String Id { get; set; }
    2 references
    public int Credit { get; set; }
    4 references
    public List<String> RegStudents { get; set; }
    2 references
    public Course(Teacher teacher, String name, String id, int credit, List<String> regStudents)
    {
        this.Teacher = teacher;
        this.Name = name;
        this.Id = id;
        this.Credit = credit;
        this.RegStudents = regStudents;
    }
}
```

Létrehozott függvények:

Setup: Egy segédfüggvény, amely feltölti az adatbázisunkat kezdő adatokkal. Az „adatbázis” jelen esetben négy lista, amelyekben elmentjük a tanárokat, tanulókat, kurzusokat és adminisztrátorokat.

```
1 reference
public static void Setup(List<Teacher> teachers, List<Student> students, List<Course> courses, List<Admin> admins)
{
    ////////// Ez egy segéd függvény, amely feltölti az adatbázist kezdő adatokkal //////////

    teachers.Add(new Teacher("Kovács Gábor", "AB12C3"));
    teachers.Add(new Teacher("Papp Laura", "XY98Z1"));

    students.Add(new Student(false, "NULL", "NULL", 0));
    students.Add(new Student(true, "Horváth Réka", "LM45T6", 0));
    students.Add(new Student(true, "Takács Levente", "PQ77B2", 0));

    courses.Add(new Course(teachers[0], "Mesterséges Intelligencia Alapjai", "MIAI", 3, new List<String>());
    courses.Add(new Course(teachers[1], "Operációs Rendszerek", "OPRE", 4, new List<String>());

    admins.Add(new Admin("NULL", "NULL"));
    admins.Add(new Admin("Németh Márk", "JK31D4"));
    admins.Add(new Admin("Tóth Ivett", "G0NBC4"));
}
```

ChangeStatus: A tanuló státuszát tudjuk vele változtatni. Megadjuk a tanuló adatbázisát, valamint a módosítani kívánt tanuló ID-jét. **foreach**-el ellenőrizzük, hogy a tanuló szerepel-e az adatbázisban és, ha igen, akkor a státuszát átváltjuk a jelenlegi ellenkezőjére. Sikeres módosítás esetén a program visszajelzést ad a felhasználónak.

```
public static void ChangeStatus(List<Student> students, String usredId)
{
    foreach (Student student in students)
    {
        if(student.Id == usredId)
        {
            student.Status = !student.Status;
            Console.WriteLine($"{student.Name} ({student.Id}) hallgató státusza megváltozott!\nÚj státusz: {student.Status}");
            Console.ReadKey();
        }
    }
}
```

ListStudents: Kिलistázza **foreach** segítségével a tanulókat adatait, az őket tartalmazó listából. Leírja a nevüket, azonosítójukat, státuszukat.

```
public static void ListStudents(List<Student> students)
{
    Console.Clear();
    foreach (Student student in students) Console.WriteLine($"Név: {student.Name} | Azonosító: {student.Id} | Státusz: {student.Status}");
    Console.WriteLine("\nA továbblépéshez nyomjon ENTER-t");
    Console.ReadKey();
}
```

ListCourses: A tanulókhöz hasonlóan **foreach** segítségével kilistázza az őket tartalmazó listából a kurzusok nevét, azonosítóját és kredit értéküket.

```
public static void ListCourses(List<Course> courses)
{
    Console.Clear();
    foreach (Course course in courses) Console.WriteLine($"Név: {course.Name} | Azonosító: {course.Id} | Kredit: {course.Credit}");
    Console.WriteLine("\nA továbblépéshez nyomjon ENTER-t");
    Console.ReadKey();
}
```

CourseEnter: Ennek a függvénynek a segítségével lehet a tanulókat jelentkeztetni a kurzusokra. A kurzusok és a tanulók listáját használva, megadjuk a felvenni kívánt kurzus négy jegyű kódját, valamint a tanuló hat jegyű azonosítóját. A függvény ellenőrzi, hogy a felvenni kívánt tanuló szerepel-e már a kurzus tagjai között és ha igen akkor vissza jelzést ad a felhasználónak erről és nem engedi újra felvenni a kurzust. Amennyiben a tanuló nem szerepel a kurzuson, felveszi annak ID-jét a regisztrált tanulók listájára. Továbbá, ha hibásan adtuk meg a tanuló ID-jét, arról is hibaüzenetet kapunk.

```
public static void CourseEnter(List<Course>courses,List<Student>students,String courseId,String userId)
{
    Boolean run = false;

    foreach (Student student in students)
    {
        if (student.Id == userId) run = true;
    }
    if (run)
    {
        foreach (Course course in courses)
        {
            if (course.Id == courseId)
            {
                if (course.RegStudents.Contains(userId))
                {
                    Console.WriteLine($"A hallgató már felvette ezt a(z) {course.Name} tárgyat!");
                    Console.ReadKey();
                    break;
                }
                else
                {
                    course.RegStudents.Add(userId);
                    Console.WriteLine($"{userId} kódú hallgató sikeresen felvette a(z) {course.Name} tárgyat!");
                    Console.ReadKey();
                }
            }
        }
    }
    if (!run)
    {
        Console.WriteLine($"HIBA! A keresett hallgató ({userId}) nem található a rendszerben!");
        Console.ReadKey();
    }
}
```

CourseQuit: A kurzusok listáját használva, megadjuk a módosítani kívánt kurzus ID-jét valamint a tanuló azonosítóját, majd az eltávolításról visszajelzést kap a felhasználó. (Jelen verzióban, ha hibás adatot adunk meg, a program lefut, viszont nem ad visszajelzést a hibáról. Azonban a program stabilitását ez nem befolyásolja)

```
public static void CourseQuit(List<Course>courses,string courseId,string userId)
{
    foreach (Course course in courses)
    {
        if (course.Id == courseId)
        {
            course.RegStudents.Remove(userId);
            Console.WriteLine($"{userId} kódú tanuló sikeresen leadta a(z) {course.Name} tárgyat! ");
            Console.ReadKey();
        }
    }
}
```

LoginStudent: Ez a függvény fut le, ha tanulóként lépünk be. Az azonosító kód megadása után a függvény kikeresi a tanulók listájából a beléptetni kívánt tanulót. Amennyiben hibás vagy nem létező kódot adtunk meg, a program visszajelzést ad erről.

Belépés után a **user** segédváltozóba lehívjuk a tanuló adatait. A program kiírja a bejelentkezett tanuló adatait (Név, ID, státusz), majd felsorolja a menü lehetőségeit. Egy **switch** elágazás tölti be a menü szerepét, amelyben a szám gombok segítségével tudunk navigálni.

```
public static void LoginStudent(List<Student>students,List<Course>courses)
{
    Boolean run = false;
    Student user = students[0];

    while (user.Name == "NULL") //Az első loop. Itt választjuk ki a használni kívánt fiókot diákként
    {
        Console.Clear();
        Console.WriteLine("Adja meg az azonosító kódját(LM45T6) vagy kilépéshez \"EXIT\": ");
        String input = Console.ReadLine();
        if (input == "EXIT") break;

        foreach (Student student in students)
        {
            if (student.Id == input)
            {
                user = student;
                run = true;
            }
        }
        if (run) break;
        Console.WriteLine("Helytelen kódot adott meg! Próbálja újra!");
    }

    while(run)
    {
        Console.Clear();
        Console.WriteLine($"Üdvözlöm {user.Name} ({user.Id}) | Státusz: {user.Status} \n");
        Console.WriteLine("Válasszon a menüből:\n1. Kilépés\n2. Státusz átváltása\n3. Kurzusra jelentkezés\n4. Kurzus leadása\n5. Kurzusok listája\n");

        switch(Console.ReadLine())
        {
            case "1": run = false; break;
            case "2": ChangeStatus(students,user.Id); break;
            case "3":
            {
                Console.WriteLine("Adja meg a felvenni kívánt kurzus 4 jegyű kódját: ");
                CourseEnter(courses,students, Console.ReadLine(), user.Id);
            } break;
            case "4":
            {
                Console.WriteLine("Adja meg a leadni kívánt kurzus 4 jegyű kódját: ");
                CourseQuit(courses, Console.ReadLine(), user.Id);
            } break;
            case "5":
                ListCourses(courses); break;

            default: break;
        }
    }
}
```

LoginAdmin: Ez a függvény hasonlóan működik a tanulók belépéséhez, viszont több opcióval rendelkezik, valamint amíg a tanulók csak saját fiókjukat módosíthatják, az adminisztrátori belépéssel bármely tanuló fiókja módosítható.

Az azonosító kód megadása után a függvény kikeresi az adminisztrátorok listájából a beléptetni kívánt személyt. Amennyiben hibás vagy nem létező kódot adtunk meg, a program visszajelzést ad erről.

Belépés után a **user** segédváltozóba lehívjuk az adminisztrátor adatait. A program kiírja a bejelentkezett adminisztrátor adatait (Név, ID), majd felsorolja a menü lehetőségeit. Egy **switch** elágazás tölti be a menü szerepét, amelyben a szám gombok segítségével tudunk navigálni.

```
public static void LoginAdmin(List<Admin>admins,List<Student>students,List<Course>courses)
{
    Boolean run = false;
    Admin user = admins[0];
    while (user.Name == "NULL") //Az első loop. Itt választjuk ki a használni kívánt fiókot diákként
    {
        Console.Clear();
        Console.WriteLine("Adj meg az azonosító kódját(G0NBC4) vagy kilépéshez \"EXIT\": ");
        String input = Console.ReadLine();
        if (input == "EXIT") break;

        foreach (Admin admin in admins)
        {
            if (admin.Id == input)
            {
                user = admin;
                run = true;
            }
        }
        if (run) break;
        Console.WriteLine("Helytelen kódot adott meg! Próbálja újra!");
    }
    while (run){
        Console.Clear();
        Console.WriteLine($"Üdvözlöm {user.Name} ({user.Id})\n");
        Console.WriteLine("Válasszon a menüből:\n1. Kilépés\n2. Hallgató státuszának váltása\n3. Hallgatók listája\n4. Hallgató kurzusra vétele");
        Console.WriteLine("5. Hallgató eltávolítása egy kurzusról\n6. Kurzusok listája");
        switch (Console.ReadLine())
        {
            case ("1"): run = false; break;
            case ("2"):
            {
                Console.WriteLine("Kérem adja meg a hallgató 6 jegyű azonosítóját: ");
                ChangeStatus(students, Console.ReadLine());
            } break;
            case ("3"): ListStudents(students); break;
            case ("4"):
            {
                Console.WriteLine("Kérem adja meg a tanuló 6 jegyű azonosítóját: ");
                String tmp = Console.ReadLine();
                Console.WriteLine("Kérem adja meg a kurzus 4 jegyű azonosítóját: ");
                CourseEnter(courses,students,Console.ReadLine(),tmp);
            } break;
            case ("5"):
            {
                Console.WriteLine("Kérem adja meg a tanuló 6 jegyű azonosítóját: ");
                String tmp = Console.ReadLine();
                Console.WriteLine("Kérem adja meg a kurzus 4 jegyű azonosítóját: ");
                CourseQuit(courses,Console.ReadLine(),tmp);
            } break;
            case ("6"): ListCourses(courses); break;
            default: break;
        }
    }
}
```

Main: A program futásakor létrehozza az „adatbázisokat”, majd a **Setup** függvény segítségével feltölti azokat adatokkal. Ezután a fő menübe kerülünk, ahol a szám gombok segítségével választhatunk, hogy tanulóként vagy adminisztrátorként szeretnénk belépni a rendszerbe.

```
0 references
static void Main(string[] args)
{
    // "Adatbázis" //////////////////////////////////////
    List<Teacher> teachers = new List<Teacher>();
    List<Student> students = new List<Student>();
    List<Admin> admins = new List<Admin>();
    List<Course> courses = new List<Course>();
    //////////////////////////////////////

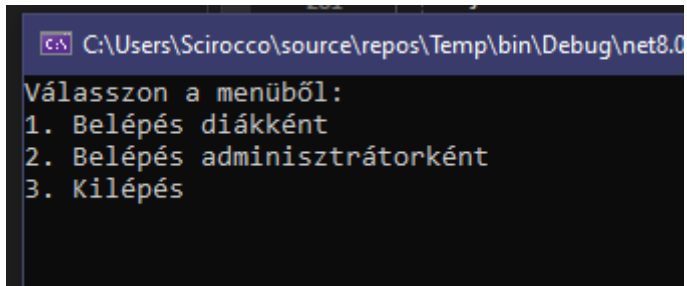
    Setup(teachers, students, courses, admins); // Feltölti alap adatokkal az "adatbázist".
    Boolean run = true;

    while (run) // Fő menü loop
    {
        Console.Clear();
        Console.WriteLine("Válasszon a menüből:\n1. Belépés diákként\n2. Belépés adminisztrátorként\n3. Kilépés");
        switch (Console.ReadLine())
        {
            case "1": LoginStudent(students, courses); break;
            case "2": LoginAdmin(admins, students, courses); break;
            case "3": run = false; break;
            default: break;
        }
    }
}
```


A program használata

Fő menü:

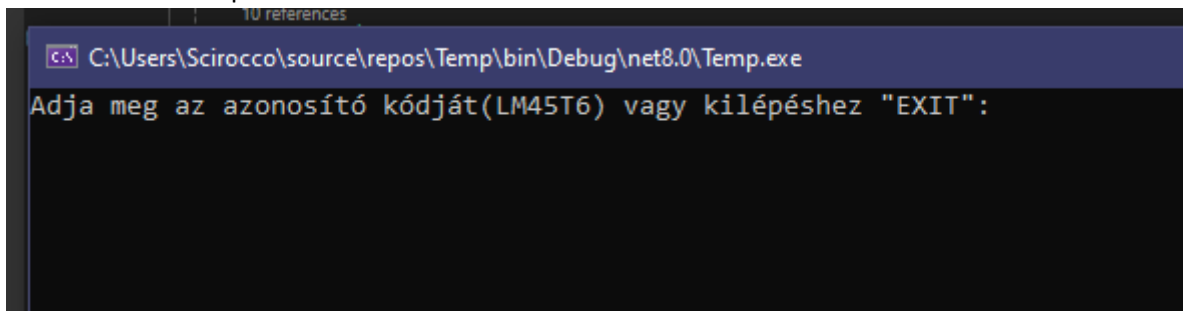
A program indítása után ebbe a menübe kerülünk. A szám gombok használatával tudunk tovább lépni vagy kilépni a programból.



```
C:\Users\Scirocco\source\repos\Temp\bin\Debug\net8.0
Válasszon a menüből:
1. Belépés diákként
2. Belépés adminisztrátorként
3. Kilépés
```

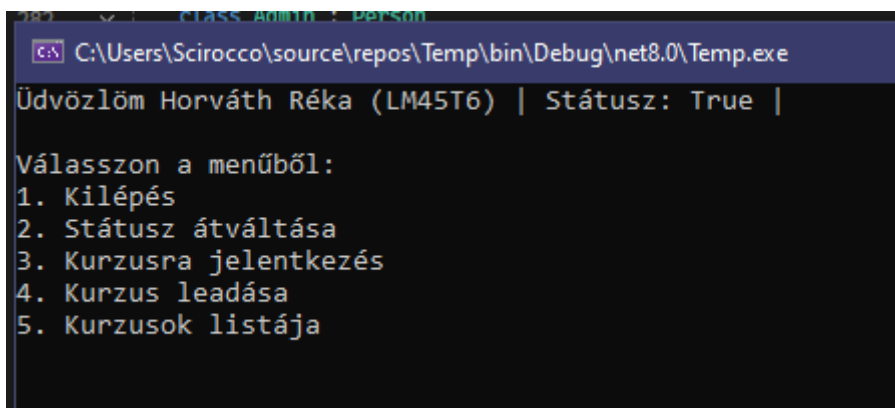
Belépés diákként:

Ahhoz, hogy a program tovább engedje a felhasználót, szükséges megadni a tanuló hat jegyű azonosítóját (Itt zárójelben megadtam egy azonosítót tesztelés céljából), vagy az „EXIT” szó beírásával vissza léphetünk a főmenübe.



```
C:\Users\Scirocco\source\repos\Temp\bin\Debug\net8.0\Temp.exe
Adja meg az azonosító kódját(LM45T6) vagy kilépéshez "EXIT":
```

Sikeres belépés után a program kiírja a tanuló adatait, majd a szám gombok segítségével navigálhatunk a menüben



```
C:\Users\Scirocco\source\repos\Temp\bin\Debug\net8.0\Temp.exe
Üdvözlöm Horváth Réka (LM45T6) | Státusz: True |
Válasszon a menüből:
1. Kilépés
2. Státusz átváltása
3. Kurzusra jelentkezés
4. Kurzus leadása
5. Kurzusok listája
```

A **státusz megváltoztatása** menüponttal, a tanuló státusza a jelenlegi ellenkezőjére vált (True/False), melyről visszajelzést kap a felhasználó.

```
C:\Users\Scirocco\source\repos\Temp\bin\Debug\net8.0\Temp.exe
Üdvözlöm Horváth Réka (LM45T6) | Státusz: False |

Válasszon a menüből:
1. Kilépés
2. Státusz átváltása
3. Kurzusra jelentkezés
4. Kursus leadása
5. Kurzusok listája

2

Horváth Réka (LM45T6) hallgató státusza megváltozott!
Új státusza: True
```

A **kurzusra jelentkezés** menüpont alatt a program bekéri a felhasználótól a felvenni kívánt kurzus négy jegyű azonosítóját (a példában MIAI). Majd visszajelzést ad amennyiben a tanuló sikeresen felvette a kurzust...

```
C:\Users\Scirocco\source\repos\Temp\bin\Debug\net8.0\Temp.exe
Üdvözlöm Horváth Réka (LM45T6) | Státusz: True |

Válasszon a menüből:
1. Kilépés
2. Státusz átváltása
3. Kurzusra jelentkezés
4. Kursus leadása
5. Kurzusok listája

3
Adja meg a felvenni kívánt kurzus 4 jegyű kódját:
MIAI
LM45T6 kódú hallgató sikeresen felvette a(z) Mesterséges Intelligencial Alapjai tárgyat!
```

...vagy már szerepel a kurzus listáján.

```
C:\Users\Scirocco\source\repos\Temp\bin\Debug\net8.0\Temp.exe
Üdvözlöm Horváth Réka (LM45T6) | Státusz: True |

Válasszon a menüből:
1. Kilépés
2. Státusz átváltása
3. Kurzusra jelentkezés
4. Kursus leadása
5. Kurzusok listája

3
Adja meg a felvenni kívánt kurzus 4 jegyű kódját:
MIAI
A hallgató már felvette ezt a(z) Mesterséges Intelligencial Alapjai tárgyat!
```

A **kurszus leadása** menüpontban a program bekéri a leadni kívánt kurzus négy jegyű kódját, majd visszajelzést ad, ha sikeresen leadta a tanuló a kurzust.

```
C:\Users\Scirocco\source\repos\Temp\bin\Debug\net8.0\Temp.exe
Üdvözlöm Horváth Réka (LM45T6) | Státusz: True |

Válasszon a menüből:
1. Kilépés
2. Státusz átváltása
3. Kurzusra jelentkezés
4. Kurszus leadása
5. Kurzusok listája

4
Adja meg a leadni kívánt kurzus 4 jegyű kódját:
MIAI
LM45T6 kódú tanuló sikeresen leadta a(z) Mesterséges Intelligencial Alapjai tárgyat!
```

A **kurzusok listája** menüpont kilistázza az összes elérhető kurzust adatait.

```
C:\Users\Scirocco\source\repos\Temp\bin\Debug\net8.0\Temp.exe
Név: Mesterséges Intelligencial Alapjai | Azonosító: MIAI | Kredit: 3
Név: Operációs Rendszerek | Azonosító: OPRE | Kredit: 4

A továbblépéshez nyomjon ENTERT
```

Belépés adminisztrátorként:

A tanuló belépéséhez hasonlóan az adminisztrátoroknak is meg kell adniuk hat jegyű azonosítójukat. (a példa kedvéért itt is megadtam zárójelben egy azonosítót)

```
C:\Users\Scirocco\source\repos\Temp\bin\Debug\net8.0\Temp.exe
Adja meg az azonosító kódját(G0NBC4) vagy kilépéshez "EXIT": _
```

Belépés után egy kibővített menüt kapunk

```
C:\Users\Scirocco\source\repos\Temp\bin\Debug\net8.0\Temp.exe
Üdvözlöm Tóth Ivett (G0NBC4)
Válasszon a menüből:
1. Kilépés
2. Hallgató státuszának váltása
3. Hallgatók listája
4. Hallgató kurzusra vétele
5. Hallgató eltávolítása egy kurzusról
6. Kurzusok listája
_
```

A **hallgató státuszának váltása** menüben szükséges megadni a hallgató hat jegyű azonosítóját. Sikeres változtatás után, a felhasználó visszajelzést kap.

```
C:\Users\Scirocco\source\repos\Temp\bin\Debug\net8.0\Temp.exe
Üdvözlöm Tóth Ivett (G0NBC4)
Válasszon a menüből:
1. Kilépés
2. Hallgató státuszának váltása
3. Hallgatók listája
4. Hallgató kurzusra vétele
5. Hallgató eltávolítása egy kurzusról
6. Kurzusok listája
2
Kérem adja meg a hallgató 6 jegyű azonosítóját:
LM45T6

Horváth Réka (LM45T6) hallgató státusza megváltozott!
Új státusza: False
```

A **Hallgató kurzusra vétele** valamint a **Hallgató eltávolítása egy kurzusról** menüben szükséges megadni a hallgató hat jegyű azonosítóját valamint a kezelni kívánt kurzus négy jegyű kódját. Sikeres változtatás után, a felhasználó visszajelzést kap.

```
C:\Users\Scirocco\source\repos\Temp\bin\Debug\net8.0\Temp.exe
Üdvözlöm Tóth Ivett (G0NBC4)
Válasszon a menüből:
1. Kilépés
2. Hallgató státuszának váltása
3. Hallgatók listája
4. Hallgató kurzusra vétele
5. Hallgató eltávolítása egy kurzusról
6. Kurzusok listája
4
Kérem adja meg a tanuló 6 jegyű azonosítóját:
LM45T6
Kérem adja meg a kurzus 4 jegyű azonosítóját:
MIAI
LM45T6 kódú hallgató sikeresen felvette a(z) Mesterséges Intelligencial Alapjai tárgyat!
```

A **Hallgatók listája** menüpont klistázza az adatbázisból a hallgatók adatait.

```
C:\Users\Scirocco\source\repos\Temp\bin\Debug\net8.0\Temp.exe
Név: NULL | Azonosító: NULL | Státusz: False
Név: Horváth Réka | Azonosító: LM45T6 | Státusz: False
Név: Takács Levente | Azonosító: PQ77B2 | Státusz: True

A továbblépéshez nyomjon ENTERT
```

A **Kurzusok listája** menüpont klistázza az adatbázisból az elérhető kurzusok adatait.

```
C:\Users\Scirocco\source\repos\Temp\bin\Debug\net8.0\Temp.exe
Név: Mesterséges Intelligencial Alapjai | Azonosító: MIAI | Kredit: 3
Név: Operációs Rendszerek | Azonosító: OPRE | Kredit: 4

A továbblépéshez nyomjon ENTERT
```