

Movie project.

Müller Zsolt István, Bartkó Milán

A program felhasználási előnyei

- A csapatunk által készített program tökéletesen megfelel egy gyakorlatban is használatra képes jegy rendszerrel amit fel lehet használni eseményeknél, mozikba illetve tökéletes színházi jegyek kezelésére is!
- A program fejlesztése során oda figyelt a csapat arra, hogy a felhasználó ne tudjon hibákat elkövetni, tehát a nevek illetve az ülések száma sem tud ismétlődni!
- A program fejlődési fázisban van. Hordozható rendszerekre is ki akarjuk adni a programot mint például egy telefon, hogy kényelmesebb legyen a dolgozóknak a munka során felvenni a mozozni vágyókat.

A program felhasználási előnyei

- Továbbá a program megfelelő az oktatásban is, a vendéglátási iskolákban a program szabadalmaztatása hatalmas előnyökkel járhat.
- Használata pofon egyszerű szinte már magától értetődő!
- Online jegyértékesítés: A program továbbfejleszthető úgy, hogy online felületen is elérhető legyen, így a felhasználók otthonról is foglalhatnak és vásárolhatnak jegyeket.
- Jegyfoglalási rendszer integrációja: A program könnyen integrálható más jegyfoglalási rendszerekbe, például weboldalakon vagy mobilalkalmazásokban.
- Ezek csak néhány példa a felhasználási formáira és előnyeire. A program könnyen testreszabható és továbbfejleszthető az adott igényeknek megfelelően.



A GitHub
oldalunk

A program

***** SZÍNHÁZI JEGY ÉRTÉKESÍTŐ PROGRAM *****

1. Nézőter megjelenítése
2. Szabad helyek kijelzése
3. Foglалás
4. Foglалás módosítása
5. Foglалás törlése
6. Mentés
7. Betöltés
8. Kilépés

Kérem, válasszon egy menüpontot (1-8): 1

***** NÉZŐTÉR MEGJELENÍTÉSE *****

```
S S S F S S S S S S S S S S
S S S S S S S S S S S S S S
F S S S S S S S S S S S S S
S F S S S S S S S S S F F S
S S F S S S S F S S S S S S
S S S S S S S S S S S S S S
F S S S S S S S S S S S S S
S S F S S S F S S S S S S S
S S S S S S F S S S S S S S
S S F S S S S S S F S S S F
S S S S S S S S S S F S S S
S S S S S S F S F F S S S S
S S S F S S S S S F S S S F
S S F S S S S S S S S S S S
S S S S S S S S S S S F S F
S S S S S S S S S S S F S S
```

Nyomjon meg egy gombot a folytatáshoz...

A program

```
Szabad hely: Sor 13, Szék 7  
Szabad hely: Sor 13, Szék 8  
Szabad hely: Sor 13, Szék 9  
Szabad hely: Sor 13, Szék 10  
Szabad hely: Sor 13, Szék 12  
Szabad hely: Sor 13, Szék 13  
Szabad hely: Sor 13, Szék 14  
Szabad hely: Sor 14, Szék 1  
Szabad hely: Sor 14, Szék 2  
Szabad hely: Sor 14, Szék 4  
Szabad hely: Sor 14, Szék 5  
Szabad hely: Sor 14, Szék 6  
Szabad hely: Sor 14, Szék 7  
Szabad hely: Sor 14, Szék 8  
Szabad hely: Sor 14, Szék 9  
Szabad hely: Sor 14, Szék 10  
Szabad hely: Sor 14, Szék 11  
Szabad hely: Sor 14, Szék 12  
Szabad hely: Sor 14, Szék 13  
Szabad hely: Sor 14, Szék 14  
Szabad hely: Sor 14, Szék 15  
Szabad hely: Sor 15, Szék 1  
Szabad hely: Sor 15, Szék 2  
Szabad hely: Sor 15, Szék 3  
Szabad hely: Sor 15, Szék 4  
Szabad hely: Sor 15, Szék 5  
Szabad hely: Sor 15, Szék 6  
Szabad hely: Sor 15, Szék 7  
Szabad hely: Sor 15, Szék 8  
Szabad hely: Sor 15, Szék 9  
Szabad hely: Sor 15, Szék 10  
Szabad hely: Sor 15, Szék 11  
Szabad hely: Sor 15, Szék 12  
Szabad hely: Sor 15, Szék 14  
Szabad hely: Sor 16, Szék 1  
Szabad hely: Sor 16, Szék 2  
Szabad hely: Sor 16, Szék 3  
Szabad hely: Sor 16, Szék 4  
Szabad hely: Sor 16, Szék 5  
Szabad hely: Sor 16, Szék 6  
Szabad hely: Sor 16, Szék 7  
Szabad hely: Sor 16, Szék 8  
Szabad hely: Sor 16, Szék 9  
Szabad hely: Sor 16, Szék 10  
Szabad hely: Sor 16, Szék 11  
Szabad hely: Sor 16, Szék 12  
Szabad hely: Sor 16, Szék 14  
Szabad hely: Sor 16, Szék 15
```

Nyomjon meg egy gombot a folytatáshoz...

A program

```
***** SZÍNHÁZI JEGY ÉRTÉKESÍTŐ PROGRAM *****
```

1. Nézőtér megjelenítése
2. Szabad helyek kijelzése
3. Foglалás
4. Foglалás módosítása
5. Foglалás törlése
6. Mentés
7. Betöltés
8. Kilépés

```
Kérem, válasszon egy menüpontot (1-8): 3
```

```
***** FOGLALÁS *****
```

```
Adja meg a nevét: Feri
```

```
Hány jegyet szeretne foglalni? 3
```

```
Válassza ki a szabad helyeket:
```

```
Hely 1: 1
```

```
Hely 2: 2
```

```
Hely 3: 3
```

```
Foglалás sikeres!
```

```
Nyomjon meg egy gombot a folytatáshoz...
```

A program

```
C:\Users\kismu\source\repos\ X + v
***** SZÍNHÁZI JEGY ÉRTÉKESÍTŐ PROGRAM *****
1. Nézőter megjelenítése
2. Szabad helyek kijelzése
3. Foglalás
4. Foglalás módosítása
5. Foglalás törlése
6. Mentés
7. Betöltés
8. Kilépés
Kérem, válasszon egy menüpontot (1-8): 4
***** FOGLALÁS MÓDOSÍTÁSA *****
Adja meg a nevét: Feri
Jelenlegi foglalásai:
- Hely 1
- Hely 2
- Hely 3
Melyik foglalást szeretné módosítani?
```


A program

***** SZÍNHÁZI JEGY ÉRTÉKESÍTŐ PROGRAM *****

1. Nézőter megjelenítése
2. Szabad helyek kijelzése
3. Foglalás
4. Foglalás módosítása
5. Foglalás törlése
6. Mentés
7. Betöltés
8. Kilépés

Kérem, válasszon egy menüpontot (1-8): 5

***** FOGLALÁS TÖRLÉSE *****

Adja meg a nevét: Feri

Nincs ilyen névvel rendelkező foglalás.

Nyomjon meg egy gombot a folytatáshoz...

A program

```
C:\Users\kismu\source\repos\ X + v
***** SZÍNHÁZI JEGY ÉRTÉKESÍTŐ PROGRAM *****
1. Nézőtér megjelenítése
2. Szabad helyek kijelzése
3. Foglалás
4. Foglалás módosítása
5. Foglалás törlése
6. Mentés
7. Betöltés
8. Kilépés
Kérem, válasszon egy menüpontot (1-8): 6
Foglалások mentve.

Nyomjon meg egy gombot a folytatáshoz...
|
```

A program

```
C:\Users\kismu\source\repos\ X + v
***** SZÍNHÁZI JEGY ÉRTÉKESÍTŐ PROGRAM *****
1. Nézőter megjelenítése
2. Szabad helyek kijelzése
3. Foglалás
4. Foglалás módosítása
5. Foglалás törlése
6. Mentés
7. Betöltés
8. Kilépés
Kérem, válasszon egy menüpontot (1-8): 7
Foglалások betöltve.

Nyomjon meg egy gombot a folytatáshoz...
```

Tabnine | 0 references

```
static void Main()
{
    RandomFoglalasokGeneralo(10);

    while (true)
    {
        Console.Clear();
        KiirMenu();
        string választas = Console.ReadLine();

        switch (választas)
        {
            case "1":
                NezetMegjelenites();
                break;
            case "2":
                SzabadHelyekKijelzese();
                break;
            case "3":
                Foglalas();
                break;
            case "4":
                FoglalasModositasa();
                break;
            case "5":
                FoglalasTorlese();
                break;
            case "6":
                Mentese();
                break;
            case "7":
                Betoltes();
                break;
            case "8":
                Environment.Exit(0);
                break;
            default:
                Console.WriteLine("Érvénytelen választás. Kérem, válasszon újra!");
                break;
        }

        Console.WriteLine("\nNyomjon meg egy gombot a folytatáshoz...");
        Console.ReadKey();
    }
}
```

Tabnine | 1 reference

```
static void KiirMenu()  
{  
    Console.WriteLine("***** SZÍNHÁZI JEGY ÉRTÉKESÍTŐ PROGRAM *****");  
    Console.WriteLine("1. Nézőter megjelenítése");  
    Console.WriteLine("2. Szabad helyek kijelzése");  
    Console.WriteLine("3. Foglалás");  
    Console.WriteLine("4. Foglалás módosítása");  
    Console.WriteLine("5. Foglалás törlése");  
    Console.WriteLine("6. Mentés");  
    Console.WriteLine("7. Betöltés");  
    Console.WriteLine("8. Kilépés");  
    Console.Write("Kérem, válasszon egy menüpontot (1-8): ");  
}
```

Tabnine | 1 reference

```
static void NezetMegjelenites()
{
    Console.WriteLine("***** NÉZŐTÉR MEGJELENÍTÉSE *****");
    for (int i = 0; i < 16; i++)
    {
        for (int j = 0; j < 15; j++)
        {
            Console.Write(nezoTere[i, j] + " ");
        }
        Console.WriteLine();
    }
}
```

laonline | I reference

```
static void SzabadHelyekKijelzese()
{
    Console.WriteLine("***** SZABAD HELYEK KIJELEZÉSE *****");
    for (int i = 0; i < 16; i++)
    {
        for (int j = 0; j < 15; j++)
        {
            if (nezoTere[i, j] == 'S')
            {
                Console.WriteLine($"Szabad hely: Sor {i + 1}, Szék {j + 1}");
            }
        }
    }
}
```

Tabnine | 1 reference

```
static void FoglalasModositasa()
{
    Console.WriteLine("***** FOGLALÁS MÓDOSÍTÁSA *****");
    Console.Write("Adja meg a nevét: ");
    string nev = Console.ReadLine();

    if (foglalasok.ContainsKey(nev))
    {
        Console.WriteLine("Jelenlegi foglalásai:");

        foreach (int hely in foglalasok[nev])
        {
            Console.WriteLine($"- Hely {hely}");
        }

        Console.Write("Melyik foglalást szeretné módosítani? ");
        int modositandoHely = int.Parse(Console.ReadLine());

        if (foglalasok[nev].Contains(modositandoHely))
        {
            nezotere[(modositandoHely - 1) / 15, (modositandoHely - 1) % 15] = 'S';
            foglalasok[nev].Remove(modositandoHely);

            Console.WriteLine("Módosítás sikeres!");
        }
        else
        {
            Console.WriteLine("Érvénytelen foglalás. Kérem, válasszon újat.");
        }
    }
    else
    {
        Console.WriteLine("Nincs ilyen névvel rendelkező foglalás.");
    }
}
```


Tabnine | 1 reference

```
static void FoglalasTorlese()
{
    Console.WriteLine("***** FOGLALÁS TÖRLÉSE *****");
    Console.Write("Adja meg a nevét: ");
    string nev = Console.ReadLine();

    if (foglalasok.ContainsKey(nev))
    {
        Console.WriteLine("Jelenlegi foglalásai:");

        foreach (int hely in foglalasok[nev])
        {
            Console.WriteLine($"- Hely {hely}");
            nezotere[(hely - 1) / 15, (hely - 1) % 15] = 'S';
        }

        foglalasok.Remove(nev);
        Console.WriteLine("Törlés sikeres!");
    }
    else
    {
        Console.WriteLine("Nincs ilyen névvel rendelkező foglalás.");
    }
}

static string mentettFajl = @"C:\Users\kismu\OneDrive\foglalasok.txt";
```

Tabnine | 1 reference

```
static void Betoltes()
{
    foglalasok.Clear();

    if (File.Exists(mentettFajl))
    {
        using (StreamReader reader = new StreamReader(mentettFajl))
        {
            string sor;
            while ((sor = reader.ReadLine()) != null)
            {
                string[] adatok = sor.Split(',');
                string nev = adatok[0];
                List<int> helyek = new List<int>();

                for (int i = 1; i < adatok.Length; i++)
                {
                    helyek.Add(int.Parse(adatok[i]));
                }

                foglalasok.Add(nev, helyek);
            }
        }
        Console.WriteLine("Foglalások betöltve.");
    }
    else
    {
        Console.WriteLine("Nem található mentett fájl.");
    }
}
```

Tabnine | 1 reference

```
static void RandomFoglalasokGeneralo(int szazalek)
{
    Random random = new Random();

    for (int i = 0; i < 16; i++)
    {
        for (int j = 0; j < 15; j++)
        {
            if (random.Next(100) < szazalek)
            {
                nezotere[i, j] = 'F';
            }
            else
            {
                nezotere[i, j] = 'S';
            }
        }
    }
}
```

Tabnine | 1 reference

```
static void Mentese()  
{  
    using (StreamWriter writer = new StreamWriter(mentettFajl))  
    {  
        foreach (var kvp in foglalasok)  
        {  
            writer.WriteLine($"{kvp.Key}, {string.Join(", ", kvp.Value)}");  
        }  
    }  
    Console.WriteLine("Foglalások mentve.");  
}
```