

1. Fájlstruktúra előkészítése
2. Berles osztály létrehozása
3. Adatok beolvasása a fájlból

```
static void Main(string[] args)
{
    var rentals = new List<Rental>();
    foreach (var line in File.ReadAllLines("RentalData2024.csv").Skip(1))
    {
        var parts = line.Split(';');

        rentals.Add(new Rental
        {
            RentalId = int.Parse(parts[0]),
            UserId = parts[1],
            JewelryId = int.Parse(parts[2]),
            JewelryName = parts[3],
            DailyPrice = int.Parse(parts[4]),
            StartDate = DateTime.Parse(parts[5]),
            EndDate = DateTime.Parse(parts[6])
        });
    }
}
```

TELJES

```
using System;
using System.Globalization;
using System.Collections.Generic;
using System.IO;
using System.Linq;

namespace jewelry_console
{
    class Rental
    {
        public int RentalId { get; set; }
        public string UserId { get; set; }
        public int JewelryId { get; set; }
        public string JewelryName { get; set; }
        public int DailyPrice { get; set; }
        public DateTime StartDate { get; set; }
        public DateTime EndDate { get; set; }
        public int Duration => (EndDate - StartDate).Days + 1;
        public int TotalPrice => DailyPrice * Duration;
    }

    internal class Program
    {
        static void Main(string[] args)
        {
            var rentals = new List<Rental>();
            foreach (var line in File.ReadAllLines("RentalData2024.csv").Skip(1))
            {
                var parts = line.Split(',');

                rentals.Add(new Rental
                {
                    RentalId = int.Parse(parts[0]),
```

```

        UserId = parts[1],
        JewelryId = int.Parse(parts[2]),
        JewelryName = parts[3],
        DailyPrice = int.Parse(parts[4]),
        StartDate = DateTime.Parse(parts[5]),
        EndDate = DateTime.Parse(parts[6])
    });
}

var mostRented = rentals
    .GroupBy(r => r.JewelryName)
    .OrderByDescending(g => g.Count())
    .First();
Console.WriteLine($"1. Legnépszerűbb ékszer: {mostRented.Key}
({mostRented.Count()} alkalom)");

int totalRevenue = rentals.Sum(r => r.TotalPrice);
Console.WriteLine($"2. Összbevétel: {totalRevenue:N0} Ft");

var topUser = rentals
    .GroupBy(r => r.UserId)
    .Select(g => new { UserId = g.Key, Total = g.Sum(r =>
r.TotalPrice) })
    .OrderByDescending(u => u.Total)
    .First();
Console.WriteLine($"3. Legtöbbet költő felhasználó: {topUser.UserId}
({topUser.Total:N0} Ft)");

string keresettEkszer = "Hanga szett";
int totalDays = rentals
    .Where(r => r.JewelryName == keresettEkszer)
    .Sum(r => r.Duration);
Console.WriteLine($"4. '{keresettEkszer}' összes bérleti napja:
{totalDays} nap");

var busiestMonth = rentals
    .GroupBy(r => r.StartDate.Month)
    .OrderByDescending(g => g.Count())
    .First().Key;
string monthName = new DateTime(2024, busiestMonth,
1).ToString("MMMM", new CultureInfo("hu-HU"));
Console.WriteLine($"5. Legforgalmasabb hónap: {monthName}");
}
}

```