

## Praktikum

### Unos korisnika foruma

Ivan Radovac

Lukas Jagodin

## **Sadržaj**

- 1. Najmanje aktivan korisnik po broju postova**
- 2. Najmanje aktivan korisnik po broju komentara**
- 3. Najaktivniji korisnik po broju postova**
- 4. Najaktivniji korisnik po broju komentara**
- 5. Pretraga korisnika po imenu**
- 6. Unos novog korisnika**
- 7. Cijeli kod**

## Najmanje aktivan korisnik po broju postova

```
static void NajmanjeAktivanKorisnikPoPostovima(List<Korisnik> korisnici)
{
    if (korisnici.Count == 0)
    {
        Console.WriteLine("Nema unesenih korisnika.");
        return;
    }

    Korisnik najmanjeAktivanPoPostovima = korisnici.OrderBy(k => k.BrojPostova).First();
    Console.WriteLine("Najmanje aktivan korisnik po broju postova:");
    Console.WriteLine(najmanjeAktivanPoPostovima);
}
```

## Najmanje aktivan korisnik po broju komentara

```
static void NajmanjeAktivanKorisnikPoKomentarima(List<Korisnik> korisnici)
{
    if (korisnici.Count == 0)
    {
        Console.WriteLine("Nema unesenih korisnika.");
        return;
    }

    Korisnik najmanjeAktivanPoKomentarima = korisnici.OrderBy(k => k.BrojKomentara).First();
    Console.WriteLine("Najmanje aktivan korisnik po broju komentara:");
    Console.WriteLine(najmanjeAktivanPoKomentarima);
}
```

## Najaktivniji korisnik po broju postova

```
static void NajaktivnijiKorisnikPoPostovima(List<Korisnik> korisnici)
{
    if (korisnici.Count == 0)
    {
        Console.WriteLine("Nema unesenih korisnika.");
        return;
    }

    Korisnik najaktivnijiPoPostovima = korisnici.OrderByDescending(k => k.BrojPostova).First();
    Console.WriteLine("Najaktivniji korisnik po broju postova:");
    Console.WriteLine(najaktivnijiPoPostovima);
}
```

## Najaktivniji korisnik po broju komentara

```
static void NajaktivnijiKorisnikPoKomentarima(List<Korisnik> korisnici)
{
    if (korisnici.Count == 0)
    {
        Console.WriteLine("Nema unesenih korisnika.");
        return;
    }

    Korisnik najaktivnijiPoKomentarima = korisnici.OrderByDescending(k =>
k.BrojKomentara).First();

    Console.WriteLine("Najaktivniji korisnik po broju komentara:");
    Console.WriteLine(najaktivnijiPoKomentarima);
}
```

## Pretraga korisnika po imenu

```
static void PretragaKorisnikaPoImenu(List<Korisnik> korisnici)
{
    Console.WriteLine("Unesite ime korisnika za pretragu:");
    string imePretraga = Console.ReadLine();

    Korisnik trazenikorisnik = korisnici.FirstOrDefault(k => k.ImePrezime.Equals(imePretraga,
StringComparison.OrdinalIgnoreCase));

    if (trazenikorisnik != null)
    {
        Console.WriteLine("Pronađeni korisnik:");
        Console.WriteLine(trazenikorisnik);
    }
    else
    {
        Console.WriteLine("Korisnik nije pronađen.");
    }
}
```

## Unos novog korisnika

```
static void UnosNovogKorisnika(List<Korisnik> korisnici)
{
    Console.WriteLine("Unesite ime i prezime korisnika:");
    string imePrezime = Console.ReadLine();

    Console.WriteLine("Unesite broj postova na forumu:");
    int brojPostova;
    while (!int.TryParse(Console.ReadLine(), out brojPostova) || brojPostova < 0)
    {
        Console.WriteLine("Molimo unesite pozitivan broj.");
    }

    Console.WriteLine("Unesite broj komentara na forumu:");
    int brojKomentara;
    while (!int.TryParse(Console.ReadLine(), out brojKomentara) || brojKomentara < 0)
    {
        Console.WriteLine("Molimo unesite pozitivan broj.");
    }

    // Dodavanje novog korisnika u listu korisnika
    korisnici.Add(new Korisnik { ImePrezime = imePrezime, BrojPostova = brojPostova,
    BrojKomentara = brojKomentara });

    Console.WriteLine("Korisnik uspješno dodan.");
}
```



```
}
```

## Cijeli kod

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

class Program
{
    static void Main()
    {
        // Učitavanje korisnika iz datoteke "korisnici.txt" pri pokretanju programa
        List<Korisnik> korisnici = UcitajKorisnike("korisnici.txt");

        while (true)
        {
            // Ispis izbornika
            Console.WriteLine("Odaberite opciju:");
            Console.WriteLine("1. Najmanje aktivan korisnik po broju postova");
            Console.WriteLine("2. Najmanje aktivan korisnik po broju komentara");
            Console.WriteLine("3. Najaktivniji korisnik po broju postova");
            Console.WriteLine("4. Najaktivniji korisnik po broju komentara");
            Console.WriteLine("5. Pretraga korisnika po imenu");
            Console.WriteLine("6. Unos novog korisnika");
            Console.WriteLine("7. Izlaz");
```

```
int opcija;

// Provjera ispravnosti unosa opcije
while (!int.TryParse(Console.ReadLine(), out opcija) || opcija < 1 || opcija > 7)
{
    Console.WriteLine("Molimo unesite broj 1, 2, 3, 4, 5, 6 ili 7.");
}

if (opcija == 1)
{
    NajmanjeAktivanKorisnikPoPostovima(korisnici);
}
else if (opcija == 2)
{
    NajmanjeAktivanKorisnikPoKomentarima(korisnici);
}
else if (opcija == 3)
{
    NajaktivnijiKorisnikPoPostovima(korisnici);
}
else if (opcija == 4)
{
    NajaktivnijiKorisnikPoKomentarima(korisnici);
}
else if (opcija == 5)
{
    PretragaKorisnikaPoMenu(korisnici);
}
else if (opcija == 6)
{
    UnosNovogKorisnika(korisnici);
}
```

```

        PohraniKorisnike("korisnici.txt", korisnici);
    }
    else if (opcija == 7)
    {
        break;
    }
}
}

```

// Metoda za čitanje korisnika iz datoteke

```

static List<Korisnik> UcitajKorisnike(string nazivDatoteke)
{
    List<Korisnik> korisnici = new List<Korisnik>();

    if (File.Exists(nazivDatoteke))
    {
        string[] linije = File.ReadAllLines(nazivDatoteke);
        foreach (string linija in linije)
        {
            string[] podaci = linija.Split(';');

            if (podaci.Length == 3 && int.TryParse(podaci[1], out int postovi) && int.TryParse(podaci[2],
out int komentari))
            {
                korisnici.Add(new Korisnik { ImePrezime = podaci[0], BrojPostova = postovi,
BrojKomentara = komentari });
            }
        }
    }

    return korisnici;
}

```

```
// Metoda za pohranu korisnika u datoteku

static void PohraniKorisnike(string nazivDatoteke, List<Korisnik> korisnici)
{
    using (StreamWriter sw = new StreamWriter(nazivDatoteke))
    {
        foreach (Korisnik korisnik in korisnici)
        {
            sw.WriteLine($"{korisnik.ImePrezime};{korisnik.BrojPostova};{korisnik.BrojKomentara}");
        }
    }
}
```

```
// Metoda za unos novog korisnika

static void UnosNovogKorisnika(List<Korisnik> korisnici)
{
    Console.WriteLine("Unesite ime i prezime korisnika:");
    string imePrezime = Console.ReadLine();

    Console.WriteLine("Unesite broj postova na forumu:");
    int brojPostova;
    while (!int.TryParse(Console.ReadLine(), out brojPostova) || brojPostova < 0)
    {
        Console.WriteLine("Molimo unesite pozitivan broj.");
    }

    Console.WriteLine("Unesite broj komentara na forumu:");
    int brojKomentara;
    while (!int.TryParse(Console.ReadLine(), out brojKomentara) || brojKomentara < 0)
    {
        Console.WriteLine("Molimo unesite pozitivan broj.");
    }
}
```

```
// Dodavanje novog korisnika u listu korisnika

korisnici.Add(new Korisnik { ImePrezime = imePrezime, BrojPostova = brojPostova,
BrojKomentara = brojKomentara });

Console.WriteLine("Korisnik uspješno dodan.");
}
```

```
// Metoda za pronalazak najmanje aktivnog korisnika po broju postova
static void NajmanjeAktivanKorisnikPoPostovima(List<Korisnik> korisnici)
{
    if (korisnici.Count == 0)
    {
        Console.WriteLine("Nema unesenih korisnika.");
        return;
    }
}
```

```
Korisnik najmanjeAktivanPoPostovima = korisnici.OrderBy(k => k.BrojPostova).First();
Console.WriteLine("Najmanje aktivan korisnik po broju postova:");
Console.WriteLine(najmanjeAktivanPoPostovima);
}
```

```
// Metoda za pronalazak najmanje aktivnog korisnika po broju komentara
static void NajmanjeAktivanKorisnikPoKomentarima(List<Korisnik> korisnici)
{
    if (korisnici.Count == 0)
    {
        Console.WriteLine("Nema unesenih korisnika.");
        return;
    }
}
```

```
Korisnik najmanjeAktivanPoKomentarima = korisnici.OrderBy(k => k.BrojKomentara).First();
```

```
Console.WriteLine("Najmanje aktivan korisnik po broju komentara:");  
Console.WriteLine(najmanjeAktivanPoKomentarima);  
}
```

```
// Metoda za pronalazak najaktivnijeg korisnika po broju postova  
static void NajaktivnijiKorisnikPoPostovima(List<Korisnik> korisnici)
```

```
{  
    if (korisnici.Count == 0)  
    {  
        Console.WriteLine("Nema unesenih korisnika.");  
        return;  
    }  
}
```

```
Korisnik najaktivnijiPoPostovima = korisnici.OrderByDescending(k => k.BrojPostova).First();  
Console.WriteLine("Najaktivniji korisnik po broju postova:");  
Console.WriteLine(najaktivnijiPoPostovima);  
}
```

```
// Metoda za pronalazak najaktivnijeg korisnika po broju komentara  
static void NajaktivnijiKorisnikPoKomentarima(List<Korisnik> korisnici)
```

```
{  
    if (korisnici.Count == 0)  
    {  
        Console.WriteLine("Nema unesenih korisnika.");  
        return;  
    }  
}
```

```
Korisnik najaktivnijiPoKomentarima = korisnici.OrderByDescending(k =>  
k.BrojKomentara).First();  
Console.WriteLine("Najaktivniji korisnik po broju komentara:");  
Console.WriteLine(najaktivnijiPoKomentarima);
```

```
}
```

```
// Metoda za pretragu korisnika po imenu
```

```
static void PretragaKorisnikaPoImenu(List<Korisnik> korisnici)
```

```
{
```

```
    Console.WriteLine("Unesite ime korisnika za pretragu:");
```

```
    string imePretraga = Console.ReadLine();
```

```
    Korisnik trazeniKorisnik = korisnici.FirstOrDefault(k => k.ImePrezime.Equals(imePretraga,  
StringComparison.OrdinalIgnoreCase));
```

```
    if (trazeniKorisnik != null)
```

```
    {
```

```
        Console.WriteLine("Pronađeni korisnik:");
```

```
        Console.WriteLine(trazeniKorisnik);
```

```
    }
```

```
    else
```

```
    {
```

```
        Console.WriteLine("Korisnik nije pronađen.");
```

```
    }
```

```
}
```

```
// Klasa koja predstavlja informacije o korisniku
```

```
class Korisnik
```

```
{
```

```
    public string ImePrezime { get; set; }
```

```
    public int BrojPostova { get; set; }
```

```
    public int BrojKomentara { get; set; }
```

```
    public override string ToString()
```

```
{  
    return "${ImePrezime} - Postovi: {BrojPostova}, Komentari: {BrojKomentara}";  
}  
}  
}
```