

K1

C#

Napisati interfejs **Biljka** koji ima metodu:

- **bool Zalivanje()**.

Napisati klasu **Ruza** koja implementira interfejs **Biljka** i ima polja **naziv** (tipa string) i **vrsta** (tipa string), **starost** (tipa int), **zalivena** (tipa bool). Realizovati prazan konstruktor, konstruktor sa parametrima, get i set property za sva polja i metodu iz interfejsa **Biljka**, redefinisano metodu **ToString**.

Iz klase **Ruza** izvesti klasu **Hrizantema** koja ima i polja: **sirinaLatice** (tipa double), **boja** (tipa string), **prosecanBrojCvetova** (tipa int). Realizovati get i set property za sva polja, prazan konstruktor, konstruktor sa parametrima, redefinisane metode **Equals** i **ToString**.

Napisati klasu **Basta** koja u sebi ima **listu hrizantema**. Realizovati prazan konstruktor.

Realizovati metode:

- **void UcitajHrizanteme(string)** koja sa zadate putanje iz tekstualne datoteke učitava i dodaje hrizanteme u baštu.
- **void Sortiraj()** koja sortira hrizanteme po starosti i prosečnom broju cvetova. Ovako sortirane hrizanteme sačuvati u tekstualnoj datoteci Uredjene.txt i u binarnu datoteku garden.dat. Hrizanteme sortirati u opadajućem redosledu, prvo hrizanteme sa najmanjom starosti, a ako su starosti iste onda gledati prosečan broj cvetova (da bude u opadajućem redosledu).

-redefinisati metodu **ToString** tako da ispisuje sve informacije o bašti i svim hrizantemama.

Podaci o hrizantemama se u datoteci nalaze u sledećem formatu:
Naziv-Vrsta-Starost-Da/Ne-SirinaLatice-Boja-ProsecanBrojCvetova

Realizovati sve metode.

Napisati kratak test program.

OBAVEZNO istestirati SVE metode i SVE konstruktore.

NAPOMENE

Sortirane hrizanteme sačuvati u datoteci Uredjene.txt

Dat je primer ulazne datoteke Hrizanteme.txt

Sve što nije testirano ne donosi bodove.

Program se mora uspešno kompajlirati, u suprotnom možete dobiti najviše jedan bod.

REŠENJE

Biljka.cs

```
namespace Basta
{
    interface Biljka {
        bool Zalivanje();
    }
}
```

Ruza.cs

```
using System;

namespace Basta
{
    [Serializable]
    class Ruza : Biljka
    {
        private string naziv;
        private string vrsta;
        private int starost;
        private bool zalivena;

        public Ruza() {
            Naziv = "/";
            Vrsta = "/";
            Starost = 0;
            Zalivena = false;
        }

        public Ruza(string naziv, string vrsta, int starost, bool zalivena) {
            Naziv = naziv;
            Vrsta = vrsta;
            Starost = starost;
            Zalivena = zalivena;
        }

        public bool Zalivanje() {
            if (Zalivena)
                return false;
            else {
                Zalivena = true;
                return true;
            }
        }

        public string Naziv { get => naziv; set => naziv = value; }
        public string Vrsta { get => vrsta; set => vrsta = value; }
        public int Starost { get => starost; set => starost = value; }
        public bool Zalivena { get => zalivena; set => zalivena = value; }

        public override string ToString()
        {
            return "----- RUZA -----\n\tNaziv: " + Naziv +
                "\n\tVrsta: " + Vrsta + "\n\tStarost: " + Starost + " meseci\n\t" +
                "Zalivena: " + (Zalivena == true ? "DA" : "NE") + "\n";
        }
    }
}
```

Hzirantema.cs

```
using System;

namespace Basta
{
    [Serializable]
    class Hzirantema : Ruza
    {
        private double sirinalatice;
        private string boja;
        private int prosecanBrojCvetova;

        public Hzirantema() : base()
        {
            Sirinalatice = 0;
            Boja = "/";
            ProsecanBrojCvetova = 0;
        }

        public Hzirantema(string naziv, string vrsta, int starost, bool zalivena,
            double sirinalatice, string boja, int prosecanBrojCvetova)
            : base(naziv, vrsta, starost, zalivena)
        {
            Sirinalatice = sirinalatice;
            Boja = boja;
            ProsecanBrojCvetova = prosecanBrojCvetova;
        }

        public double Sirinalatice { get => sirinalatice; set => sirinalatice = value; }
        public string Boja { get => boja; set => boja = value; }
        public int ProsecanBrojCvetova
        {
            get => prosecanBrojCvetova;
            set => prosecanBrojCvetova = value;
        }

        public override bool Equals(object obj) {
            return obj is Hzirantema hrizantema &&
                Naziv == hrizantema.Naziv &&
                Vrsta == hrizantema.Vrsta &&
                Starost == hrizantema.Starost &&
                Zalivena == hrizantema.Zalivena &&
                Sirinalatice == hrizantema.Sirinalatice &&
                Boja == hrizantema.Boja &&
                ProsecanBrojCvetova == hrizantema.ProsecanBrojCvetova;
        }

        public override string ToString() {
            return "----- HRIZANTEMA -----\n\tNaziv: " + Naziv +
                "\n\tVrsta: " + Vrsta + "\n\tStarost: " + Starost + " meseci\n\t" +
                "Zalivena: " + (Zalivena == true ? "DA" : "NE") +
                "\n\tSirinalatice: " + Sirinalatice + "cm\n\tBoja: " + Boja +
                "\n\tProsecan Broj Cvetova: " + ProsecanBrojCvetova + " cvetova\n";
        }

        public override int GetHashCode() {
            return base.GetHashCode();
        }
    }
}
```

Basta.cs

```
using System.IO;
using System.Collections.Generic;
using System;
using System.Runtime.Serialization.Formatters.Binary;

namespace Basta
{
    class Basta
    {
        private List<Hrizantema> hrizanteme;

        public Basta()
        {
            hrizanteme = new List<Hrizantema>();
        }

        public void UcitajHrizanteme(string putanja)
        {
            TextReader tr = null;

            try
            {
                tr = new StreamReader(File.Open(putanja, FileMode.Open));

                while(true)
                {
                    string linija = tr.ReadLine();

                    if (linija == null)
                        break;
                    else
                    {
                        string[] delovi = linija.Split('-');
                        bool zaliveno;

                        if (delovi[3].ToLower() == "da")
                            zaliveno = true;
                        else
                            zaliveno = false;

                        Hrizantema tmp = new Hrizantema(delovi[0],
                                                            delovi[1],
                                                            Int32.Parse(delovi[2]),
                                                            zaliveno,
                                                            Double.Parse(delovi[4]),
                                                            delovi[5],
                                                            Int32.Parse(delovi[6]));

                        hrizanteme.Add(tmp);
                    }
                }
            }
            catch(Exception e) {
                Console.WriteLine("\n" + e.Message);
            }
            finally {
                if (tr != null)
                    tr.Close();
            }
        }
    }
}
```

```

public void Sortiraj()
{
    // IMA SMISLA SORTIRATI AKO JE VISE OD 1 HRIZANTEME U BASTI
    if(hrizanteme.Count > 1)
    {
        Hrizantema h1, h2;

        for(int i = 0; i < hrizanteme.Count - 1; i++)
            for(int j = i + 1; j < hrizanteme.Count; j++)
                if(hrizanteme[i].Starost > hrizanteme[j].Starost)
                    zameni(out h1, out h2, i, j);
                else if( (hrizanteme[i].Starost == hrizanteme[j].Starost)
                    && (hrizanteme[i].ProsecanBrojCvetova <
                        hrizanteme[j].ProsecanBrojCvetova) )
                    zameni(out h1, out h2, i, j);
    }
    SacuvajUredjeneTekstualna();
    SacuvajUredjeneBinarna();
}

private void zameni(out Hrizantema h1, out Hrizantema h2, int i, int j)
{
    h1 = hrizanteme[i];
    h2 = hrizanteme[j];

    hrizanteme.RemoveAt(i);
    hrizanteme.Insert(i, h2);

    hrizanteme.RemoveAt(j);
    hrizanteme.Insert(j, h1);
}

public void SacuvajUredjeneTekstualna(string putanja = "Uredjene.txt")
{
    TextWriter tw = null;

    try
    {
        tw = new StreamWriter(File.Open(putanja, FileMode.Create));
        string str = "";

        foreach (Hrizantema h in hrizanteme)
            str += h.ToString() + "\n";

        tw.Write(str);
    }
    catch(Exception e)
    {
        Console.WriteLine("\n" + e.Message);
    }
    finally
    {
        if (tw != null)
            tw.Close();
    }
}

```

```

public void SacuvajUredjeneBinarna(string putanja = "garden.dat")
{
    FileStream bw = null;

    try
    {
        bw = new FileStream(putanja, FileMode.Create);
        BinaryFormatter bf = new BinaryFormatter();

        try { bf.Serialize(bw, hrizanteme); }
        catch (Exception e) { Console.WriteLine("\n" + e.Message); }
    }
    catch (Exception e)
    {
        Console.WriteLine("\n" + e.Message);
    }
    finally
    {
        if (bw != null)
            bw.Close();
    }
}

public override string ToString()
{
    string str = "";
    str += "\n----- BASTA ----- \n\n";

    foreach (Hrizantema h in hrizanteme)
        str += h.ToString() + "\n";

    if (hrizanteme.Count == 0)
        str += "\n\t\t BASTA JE PRAZNA! \n\n";

    str += "\n----- \n";

    return str;
}
}

```

Program.cs

```
using System;

namespace Basta
{
    class Program
    {
        static void Main(string[] args)
        {
            // KLASA RUZA
            Ruza r1 = new Ruza();
            Ruza r2 = new Ruza("Nocna ruza", "Huolda", 9, false);

            Console.WriteLine(r1.ToString()+ "\n" + r2.ToString());

            // METODA ZALIVANJE
            if(r2.Zalivanje())
                Console.WriteLine("Ruza je zalivena!".ToUpper());
            else
                Console.WriteLine("Ruza nije zalivena!\n".ToUpper());

            // NECE SE ZALITI VEC ZALIVENA RUZA
            if (r2.Zalivanje())
                Console.WriteLine("Ruza je zalivena!".ToUpper());
            else
                Console.WriteLine("Ruza nije zalivena!\n".ToUpper());

            // KLASA HRIZANTEMA
            Hrizantema h1 = new Hrizantema();
            Hrizantema h2 = new Hrizantema("Dnevna Hrizantema", "Zbun",
                                           42, false, 14, "Plava", 24);
            Hrizantema h3 = new Hrizantema();
            Console.WriteLine(h1.ToString()+"\n" + h2.ToString() + "\n" + h3.ToString());

            // METODA EQUALS
            if(h1.Equals(h2))
                Console.WriteLine("HRIZANTEME SU ISTE!");
            else
                Console.WriteLine("HRIZANTEME NISU ISTE!");

            // METODA EQUALS
            if (h1.Equals(h3))
                Console.WriteLine("HRIZANTEME SU ISTE!");
            else
                Console.WriteLine("HRIZANTEME NISU ISTE!");

            // BASTA
            Basta b = new Basta();
            Console.WriteLine(b);

            // METODA UCITAJ I DODAJ
            b.UcitajHrizanteme("hrizanteme.txt");

            // METODA SORTIRAJ
            b.Sortiraj();
            Console.WriteLine(b);
        }
    }
}
```