# Laborator02

## Clase Generice

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication12
{
    class Multime<T>
    {
        static int dimMax = 100;
        T[] v;
        int length; //numarul de elemente

        public Multime() // constructor implicit
        {
            v = new T[dimMax];
            length = 0;
        }

        public Multime(Multime<T> M) //constructor de copiere
        {
            v = new T[dimMax];
            for (int i = 0; i < M.length; i++)
                v[i] = M.v[i];
            this.length = M.length;
        }

        public Multime(T[] v, int n) // constructor de initializare
        {
            this.v = new T[dimMax];
            int min = (v.Length < n) ? v.Length : n;
            min = (min < dimMax) ? min : dimMax;
            for (int i = 0; i < min; i++)
                this.v[i] = v[i];
            length = min;
        }

        //Accesori
        public int Length
        {
            get { return length; }
            set { length = value; }
        }

        public static int DimMax
        {
            get { return dimMax; }
            set { dimMax = value; }
        }
        //Iterator
        //Multime M, M[i] -> v[i]
        public T this[int i]
        {
            get { return v[i]; }
            set { v[i] = value; }
        }

        public bool Exista(T x)
        {
            for (int i = 0; i < length; i++)
                if (v[i].Equals(x))    //bool Equals(object o) - este mostenita
din clasa object
                                       //trebuie suprascrisa (override) in
(clasa) tipul T
                    return true;
            return false;
        }

        public bool Full()
        {
            return length == dimMax;
        }

        public bool Empty()
        {
            return length == 0;
        }

        public void Add(T x)
        {
            if (!Exista(x))
                //{
                //   v[length] = x;
                //  length++;
                // }
                v[length++] = x;

            // v[0],...., v[length-1],x
        }

        public void Delete(T x)
        {
            for (int i = 0; i < length; i++)
                if (v[i].Equals(x))
                {
                    v[i] = v[length - 1];
                    length--;
                }
            //1 2 3 4 5 6 7 8 9
            //1 2 9 4 5 6 7 8 9
        }

        public override string ToString()
        //{ 1, 2, 3, 4}
        {
            string s = "{";
            for (int i = 0; i < length; i++)
                s += v[i] + ", ";  //apel implicit v[i].ToString()
                                   // String ToString(); este mostenita din clasa
object
                                   //returneaza string
                                   //trebuie suprascrisa (override) in (clasa)
tipul T
                                   //operatorul + reprezinta aici concatenare de
siruri
            s += "\b\b}";
            return s;
        }

        //supraincarcarea operatorilor
        //C=A+B, reuniunea dintre A si B
        public static Multime<T> operator +(Multime<T> A, Multime<T> B)
        {
            Multime<T> C = new Multime<T>(A); // apel constructor de copiere
            for (int i = 0; i < B.length; i++)
                if (!A.Exista(B[i]))   //iterator  B.v[i]=B[i]
                    C.Add(B[i]);
            return C;
        }
    }

        // operator*   intersectia dintre A si B
        // operator-   A-B
    class Program
    {
        static void Main(string[] args)
        {
            int[] a = { 10, 20, 30, 40, 50 };
            Multime<int> A = new Multime<int>(a, 5);  // constructorul de
initializare
            int[] b = { 100, 200, 303, 400 };
            Multime<int> B = new Multime<int>(b, 7);
            Multime<int> C = A + B;
            Console.WriteLine("{0} + {1} = {2}", A, B, C); // apel automat al
metodei ToString

            Multime<string> persoane = new Multime<string>();
            persoane.Add("popescu ion");
            persoane.Add("cristina tudose");
            Console.WriteLine(persoane);

            //Multime<NrComplexe> multime = new Multime<NrComplexe>();
            //multime.Add(new NrComplexe(2,3));

            //Console.WriteLine("{0}", C[2]);  //iterator C.v[2]
            //Console.WriteLine("{0}", A.ToString());

            //de apelat toate metodele definite in clasa Multime
            Console.ReadKey();

        }
    }
}
```