|  |  |
| --- | --- |
| **8. Клас** “одновимірний масив” – TArray | |
| **поля** | * для зберігання елементів масиву; * для зберігання кількості елементів. |
| **методи** | * конструктор без параметрів, конструктор з параметрами, конструктор копіювання; * введення/виведення даних; * знаходження найбільшого/найменшого елемента; * сортування масиву; * знаходження суми елементів; * перевантаження операторів + (додавання елементів), – (віднімання елементів), \* (множення масиву на число). |

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace my\_project

{

    internal class Program

    {

        static void Main(string[] args)

        {

            TArray a = new TArray(7);

            TArray b = new TArray();

            Console.WriteLine($" = a{a}");

            Console.WriteLine($"min={a.Min()}");

            Console.WriteLine($"max={a.Max()}");

            Console.WriteLine($"sum={a.Sum()}");

            Console.WriteLine($" = b{b}");

            Console.WriteLine($"min={b.Min()}");

            Console.WriteLine($"max={b.Max()}");

            Console.WriteLine($"sum={b.Sum()}");

            Console.WriteLine($" = a+b{ a + b}");

            Console.WriteLine($" = a-b{ a - b}");

            Console.WriteLine($" = a\*10{ a \* 10}");

            Console.WriteLine($" <- a.sort{a.Sort()}");

            Console.ReadLine();

        }

    }

    public class TArray

    {

        private int \_length;

        public int Length { get { return \_length; } set { if (value > 0) \_length = value; else throw new Exception("Value is incorrect"); } }

        public int[] Array { get; set; }

        public TArray(int length = 3)

        {

            Random rnd = new Random();

            Length = length;

            Array = new int[Length];

            for (int i = 0; i < Array.GetLength(0); i++)

                Array[i] = rnd.Next(-20, 21);

        }

        public TArray()

        {

            Random rnd = new Random();

            Console.Write("Кількість елементів: ");

            Length = Convert.ToInt32(Console.ReadLine());

            Array = new int[Length];

            for (int i = 0; i < Array.GetLength(0); i++)

                Array[i] = rnd.Next(-20, 21);

        }

        public TArray(TArray m)

        {

            Length = m.Length;

            Array = m.Array;

        }

        public override string ToString()

        {

            Console.Write('[');

            for (int i = 0; i < Array.GetLength(0); i++)

            {

                Console.Write(Array[i]);

                Console.Write(" ");

            }

            Console.Write(']');

            return "";

        }

        public int Max()

        {

            int max = -20;

            for (int i = 0; i < Array.GetLength(0); i++)

            {

                if (Array[i] > max)

                    max = Array[i];

            }

            return max;

        }

        public int Min()

        {

            int min = 21;

            for (int i = 0; i < Array.GetLength(0); i++)

            {

                if (Array[i] < min)

                    min = Array[i];

            }

            return min;

        }

        public string Sort()

        {

            int[] array = Array;

            int x = 0;

            for (int i = 0; i < Length; i++)

                for (int j = 0; j < Length - 1 - i; j++)

                    if (array[j] > array[j + 1])

                    {

                        x = array[j];

                        array[j] = array[j + 1];

                        array[j + 1] = x;

                    }

            Console.Write('[');

            for (int i = 0; i < Length; i++)

                Console.Write($"{array[i]} ");

            Console.Write(']');

            return "";

        }

        public int Sum()

        {

            int sum = 0;

            for (int i = 0; i < Array.GetLength(0); i++)

                sum += Array[i];

            return sum;

        }

        public static TArray operator +(TArray a1, TArray a2)

        {

            int l = 0;

            if (a1.Length <= a2.Length)

                l = a1.Length;

            else

                l = a2.Length;

            TArray a = new TArray(l);

            for (int i = 0; i < l; i++)

                a.Array[i] = a1.Array[i] + a2.Array[i];

            return new TArray(a);

        }

        public static TArray operator -(TArray a1, TArray a2)

        {

            int l = 0;

            if (a1.Length <= a2.Length)

                l = a1.Length;

            else

                l = a2.Length;

            TArray a = new TArray(l);

            for (int i = 0; i < l; i++)

                a.Array[i] = a1.Array[i] - a2.Array[i];

            return new TArray(a);

        }

        public static TArray operator \*(TArray a1, int x)

        {

            TArray a = new TArray(a1.Length);

            for (int i = 0; i < a1.Length; i++)

                a.Array[i] = a1.Array[i] \* x;

            return new TArray(a);

        }

    }

}