**LINQ** (using System.Linq;)

**СЕРИАЛИЗАЦИЯ/ДЕСЕРИАЛИЗАЦИЯ**

Не забыть написать атрибут [Serializable] перед сериализуемым классом

SOAP: using System.Runtime.Serialization.Formatters.Soap;

Test test1 = new Test();

Test test2 = new Test();

Test test3 = new Test();

Test test4 = new Test();

Test[] arr\_test = { test1, test2, test3, test4};

SoapFormatter soap = new SoapFormatter();

using (FileStream fs = new FileStream("test.soap", FileMode.OpenOrCreate))

{

soap.Serialize(fs, arr\_test);

}

using (FileStream fs = new FileStream("test.soap", FileMode.OpenOrCreate))

{

soap.Deserialize(fs);

}

Binary: using System.Runtime.Serialization.Formatters.Binary;

То же самое, как в SOAP, только пишется “BinaryFormatter”

XML: using System.Xml.Serialization;

XmlSerializer xml = new XmlSerializer(typeof(Test));

А дальше все так же, как и в SOAP.

JSON: using Newtonsoft.Json;

1)

Test test1 = new Test();

Test test2 = new Test();

Test test3 = new Test();

Test test4 = new Test();

Test[] arr\_test = { test1, test2, test3, test4 };

var json = JsonSerializer.Create();

using (StreamWriter sw = new StreamWriter("./json.json", false, System.Text.Encoding.Default))

{

json.Serialize(sw, arr\_test);

}

2)

Test test1 = new Test();

Test test2 = new Test();

Test test3 = new Test();

Test test4 = new Test();

Test[] arr\_test = { test1, test2, test3, test4 };

string json = JsonConvert.SerializeObject(arr\_test); //сериализуем в строку с кодом JSON

Test[] arr\_test2 = JsonConvert.DeserializeObject<Test[]>(json);

**МАССИВЫ**

Зубчатый массив

int[][] arr2 = new int[3][];

arr2[0] = new int[2];

arr2[1] = new int[3];

arr2[2] = new int[4];

Random random = new Random();

for(int x = 0; x < arr2.Length; x++)

{

for(int y = 0; y < arr2[x].Length; y++)

{

arr2[x][y] = random.Next(0,10);

}

}

Прямоугольный массив

int[,] arr = { { 1, 2, 3 }, { 4, 5, 6 } };

for (int x = 0; x < arr.GetLength(0); x++)

{

for(int y = 0; y < arr.GetLength(1);y++)

{

Console.Write($"{arr[x, y]} ");

}

Console.WriteLine();

}

Одномерный массив

int[] arr1 = { };

int[] arr2 = new int[] { };

**ДЕЛЕГАТЫ + СОБЫТИЯ**

public delegate void Action();

static void Main(string[] args)

{

Button button = new Button();

button.click += button.Click;

button.click += button.Click;

button.Clicker();

}

public class Button

{

public event Action click;

public void Click()

{

Console.WriteLine("Click!");

}

public void Clicker()

{

click();

}

}

**ФАЙЛОВАЯ СИСТЕМА** (using System.IO;)

Диски

var test = DriveInfo.GetDrives();

foreach (var drive in test)

{

Console.WriteLine($"{drive.Name} free space -> {drive.TotalFreeSpace} bytes");

}

Файлы

FileInfo file = new FileInfo(path);

Console.WriteLine($"Creation time -> {file.CreationTime} \nLast change -> {file.LastWriteTime}");

Папки

DirectoryInfo dir = new DirectoryInfo(@"D:\OOP 2-1\lab 13\lab 13");

Console.WriteLine($"Creation time time of lab 13 -> {dir.CreationTime}");

**ПЕРЕОПРЕДЕЛЕНИЕ**

Переопределение операторов.

public static int operator +(Test2 el, int n)

Переопределение методов

public override string ToString()

**ПОТОКИ** (using System.Threading;)

Поток с параметрами

Thread new\_thread = new Thread(new ParameterizedThreadStart(Count));

new\_thread.Start(n);

Создание потоков

1) Thread myThread = new Thread(new ThreadStart(Count));

myThread.Start();

2) Thread myThread = new Thread(Count);

**TASKS** (using System.Threading.Tasks;)

Создание Task

1) Task task = new Task(() => Console.WriteLine("Hello Task!"));

task.Start();

2) Task task = Task.Factory.StartNew(() => Console.WriteLine("Hello Task!"));

3) Task task = Task.Run(() => Console.WriteLine("Hello Task!"));

**ПРОЦЕССЫ** (using System.Diagnostics;)

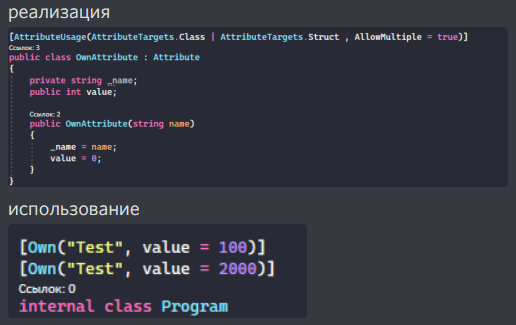
foreach (Process process in Process.GetProcesses())

{

Console.WriteLine($"ID: {process.Id} Name: {process.ProcessName}");

}

**ПОЛЬЗОВАТЕЛЬСКИЙ АТРИБУТ**



**ПЕРВЫЙ И ПОСЛЕДНИЙ ЭЛЕМЕНТ В ОТСОРТИРОВАННОМ СПИСКЕ**

