C#. Lab 7. Create Console Application. Your task is to write code which works with following program:

```
class Program
{

static void Main(string[] args)
{
        //==========Static field==============
        Console.WriteLine("Number of saved measurements:"+Measurement.NoOfMeasurements);
        //=========Inheritance================
        GammaRadiationMeasurement gammacheck1=new GammaRadiationMeasurement("CLOR GAMMA1");
        GammaRadiationMeasurement gammacheck2=new GammaRadiationMeasurement("CLOR GAMMA2");
        AirPollutionMeasurement polutioncheck1=new AirPollutionMeasurement("CLOR AIR POLLUTION");
        //==========Staticfield==============
        Console.WriteLine("Number of saved measurements:"+Measurement.NoOfMeasurements);
        //========Properties===================
        gammacheck1.RadiationLevel=0.002;
        gammacheck2.RadiationLevel=0.04;
        polutioncheck1.PM2_5=22;
        polutioncheck1.PM10=132;

        Measurement[] CLORmeasurements=new Measurement[3];
        CLORmeasurements[0]=gammacheck1;
        CLORmeasurements[1]=gammacheck2;
        CLORmeasurements[2]=polutioncheck1;

        Random rnd=new Random();
        foreach(Measurement check in CLORmeasurements)
        {
                //----------Measurementsimulation---------
                double temperature=rnd.NextDouble()*50-10;
                double presseure=rnd.Next(950,1200);
                double humidity=rnd.Next(0,100);
                //-----------------------------------------

                //===========Indexer+enum=============
                check[(int)Measurement.MeasurementConditions.Temperature]=temperature;
                check[(int)Measurement.MeasurementConditions.Pressure]=presseure;
                check[(int)Measurement.MeasurementConditions.Humidity]=humidity;
                check.Print();
        }

        //========================================================
        Console.WriteLine("\n*****HTMLPRINTING******\n");
        //============interfejsy=================================
        Report report=new Report();
        PrintToHTML(report);
        PrintToHTML(CLORmeasurements[1]);
        PrintToHTML(CLORmeasurements[2]);
        Console.ReadKey();

}

static void PrintToHTML(IGenerateHTML objectToPrint)
{
        objectToPrint.GenerateHTML();
}

}
```

The code should produce the following output:

```
Number of saved measurements:0
Number of saved measurements:3
Station name: CLOR GAMMA1, time of the measurement: 16.11.2017 11:21:47
Conditions - temperature: 3,8, pressure: 950, humidity: 26
Radiation level: 0,002
Station name: CLOR GAMMA2, time of the measurement: 16.11.2017 11:21:47
Conditions - temperature: 4,3, pressure: 1181, humidity: 16
Radiation level: 0,04
Station name: CLOR AIR POLLUTION, time of the measurement: 16.11.2017 11:21:47
Conditions - temperature: 37,7, pressure: 1160, humidity: 66
Air pollution - PM2.5: 22 PM10: 132

*****HTMLPRINTING******

<p>This is a placeholder for the raport.</p>
<font color='red' > =========POMIAR=======</font> <br />
Station name: CLOR GAMMA2, time of the measurement: 16.11.2017 11:21:47<br />
Conditions - temperature: 4,31828805446545, pressure: 1181, humidity: 16<br />

<font color='red' > =========POMIAR=======</font> <br />
Station name: CLOR AIR POLLUTION, time of the measurement: 16.11.2017 11:21:47<br />
Conditions - temperature: 37,6542711712673, pressure: 1160, humidity: 66<br />
```

Tasks (1 point each)

1. Base class, fields, enum, constructors

You should create **`Measurement`** class, with the following fields:

- **`stationName`** (type: string)
- **`date`** (type: DateTime)
- **`conditions`** (type: 1D array for 3 double) – an array which stores temperature [0], pressure [1] and humidity [2]

Constructor:

- with one parameter (name of the station – please use default value for the parameter)
  - default values are:
    - stationName: „No name!"
    - date: now (actual date)
    - conditions: 0, 0, 0

And a method:

- `void Print()` – which prints on the screen information about the measurement (date/time, temperature, pressure, and humidity)

Also:

- enum **`MeasurementConditions`** (Temperature=`0, Pressure=1, Humidity=2`)

2. Inheritance and polimorphism

Create two additional classes: `GammaRadiationMeasurement` and `AirPolutionMeasurement` which inherits from the `Measurement` class.

Fields of **`GammaRadiationMeasurement`** class:

- radiationLevel – double – direct initialization with 0

Fields of **`AirPolutionMeasurement`** class:

- PM2_5 – double – direct initialization with 0
- PM10 – double – direct initialization with 0

Common for both classes:

- should implement constructor with one parameter (name of the station), wich calls constructor from base class
- should overload **Print** method in such a way that also additional fields are printed. Base Print method should be called as well. Remember to use „**virtual**” keyword.

3. Properties and indexers

- all **fields** of classess klas `GammaRadiationMeasurement` and `AirPolutionMeasurement` should be defined as **auto-implemented properties**
- implement indexer for Measurement class:
  - for Measurement object called measurement
    - measurement[Measurement.MeasurementConditions.Temperature] – should allow to change conditions[0]
    - measurement[Measurement.MeasurementConditions.Pressure] – should allow to change conditions[1]
    - measurement[Measurement.MeasurementConditions.Humidity] – should allow to change conditions[2]
  - cast from **enum** to **int** may be useful – **(int)i**

4. Static field

- To the Measurement class additional static field should be added: **noOfMeasurements** of type *int*. Field should count number of instances of „Measurement” created in the program (you should increase value by 1 in constructor).
- Field should be associated with **read-only** property.

5. Interfaces

- Create interface „`IGenerateHTML`” which forces to implement following methods:
  - `string ToHTMLString();` – method which returns string containing information about measurement in appropriate formatting (HTML tags); see example
  - `void GenerateHTML(); -` method which writes on the screen output of ToHTMLString method.

- Add created interface into classes **Measurement**, **GammaRadiationMeasurement**, and **AirPollutionMeasurement**. Implement appropriate methods.
- Create **Raport** class, which implements created interface. **GenerateHTML** method should print out „<p>This is a placeholder for the report.</p>”.