

**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();
        }

        //=====interfejsy=====
        Console.WriteLine("\n*****HTMLPRINTING*****\n");
        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: 23.11.2020 12:38:48
Conditions - temperature: 15,4, pressure: 1056, humidity: 32
Radiation level: 0,002
Station name: CLOR GAMMA2, time of the measurement: 23.11.2020 12:38:48
Conditions - temperature: 15,8, pressure: 1142, humidity: 6
Radiation level: 0,04
Station name: CLOR AIR POLLUTION, time of the measurement: 23.11.2020 12:38:48
Conditions - temperature: -9,6, pressure: 1042, humidity: 54
Air pollution - PM2.5: 22 PM10: 132

*****HTMLPRINTING*****

<p>This is a placeholder for the report.</p>
<font color='red' > =====MEASUREMENT=====</font> <br />
Station name: CLOR GAMMA2, time of the measurement: 23.11.2020 12:38:48
Conditions - temperature: 15,8417470268168, pressure: 1142, humidity: 6<br />
Radiation level: 0,04<br />
<font color='red' > =====POMIAR=====</font> <br />
Station name: CLOR AIR POLLUTION, time of the measurement: 23.11.2020 12:38:48
Conditions - temperature: -9,61913267598447, pressure: 1042, humidity: 54<br />
Air pollution - PM2.5: 22 PM10: 132<br />
```

## Tasks

### 1. Base class, fields, enum, constructors (1.5 p)

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 „No name!“ for the parameter)
- default values for other parameters are:
  - 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 polymorphism (1.5 p)

Create two additional classes: **GammaRadiationMeasurement** and **AirPollutionMeasurement** which inherits from the **Measurement** class.

Fields of **GammaRadiationMeasurement** class:

- **radiationLevel** – double – direct initialization with 0

Fields of **AirPollutionMeasurement** 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), which 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 (1 p)

- all **fields** of classes `GammaRadiationMeasurement` and `AirPollutionMeasurement` 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)enum\_val**

### 4. Static field (1 p)

- 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 (1 p)

- 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>`”.