

AT.NET – Real-time test

Time: 90 minutes

Permitted literature: MSDN (<https://docs.microsoft.com/>) – **no other source** is allowed! You can use google for search, but only on MSDN pages (example of the query: “site:docs.microsoft.com MVC controller”)

Reflection (10 points)

Create class *SimpleXmlSerializer* with only one method called “*Serialize*”. This method will take two arguments:

1. Any object – this object will be serialized.
2. Path to the output file (XML).

Serialize input object (convert to XML string) using reflection and class *XmlDocument*, into the output file. Method must be able to serialize **any object** with base data types (string, int, double, bool).

Example class for object and related XML output:

```
0 references
public class SampleClass {
    0 references
    public string Name { get; set; }
    0 references
    public int Age { get; set; }
    0 references
    public bool IsAlive { get; set; }
}
```

```
<?xml version="1.0" encoding="utf-8" ?>
<object>
  <Name>... value ...</Name>
  <Age>... value ...</Age>
  <IsAlive>... value ...</IsAlive>
</object>
```

Web pages (10 points)

Create web application (MVC) that contains the menu with three pages.

1st page:

- Add contact form (name, email and text).
- Validate this form on the server.
- Save data from the form into the App_Data directory (the format is up to you).

2nd page:

- Add list of the saved data from the contact form (from previous step).

3rd page:

- Data from the next tasks (Downloader and XML) – print titles of the news and add links to them.

Downloader (5 points)

Create class “Downloader” - in the web application from the previous task (if you do not know how to create web application, try to complete this task using console application).

This class will contain method for downloading XML document from the URL:

<https://news.ycombinator.com/rss>

Do not save this XML on the disk – XML will be returned from the method as a text.

XML (5 points)

Use class **XmlDocument** to create list of the news and theirs links (tags **title** and **link**) from downloaded XML file.

Print the data in the 3rd page of your website (task Web Pages).