

# C# and Microsoft .NET **Serialization**



Trainer: Georgi Panayotov  
E-mail: [smg@smg-bg.net](mailto:smg@smg-bg.net)  
Phone / Viber: +359877347912

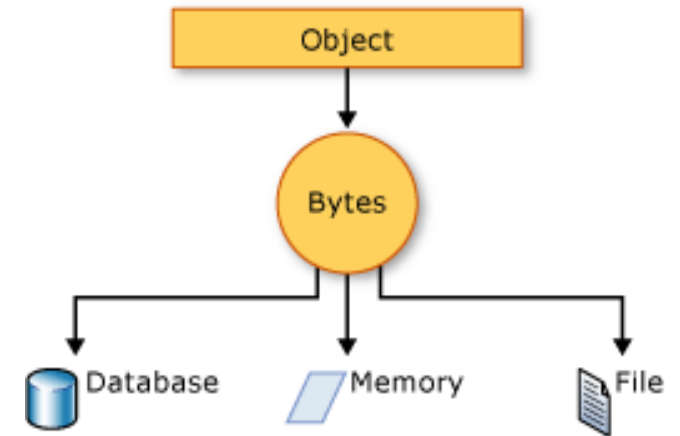


# Last time...

- Working with files
  - Stream abstract class
    - FileStream class
  - StreamWriter and StreamReader abstract classes
    - TextWriter and TextReader classes
  - Path static class
  - File and Directory static classes
  - FileInfo and DirectoryInfo classes
- Bonus lectures (just a reminder)
  - November 26<sup>th</sup> (Monday)
  - November 28<sup>th</sup> (Wednesday)
- Any questions and/or feedback?

# Serialization

*“Serialization is the process of converting an object into a stream of bytes to store the object or transmit it to memory, a database, or a file. Its main purpose is to save the state of an object in order to be able to recreate it when needed. The reverse process is called deserialization”*



[Microsoft Documentation](#)

# Binary Format

- Binary
  - Sequence of bytes
  - Platform and language dependent
  - Difficult to maintain and version
  - Efficient and performant

0 1 1 0 0  
1 0 1 1 0  
1 1 1 1 0

# Text Formats

- XML (eXtensible Markup Language)

- Text format that describes data with tags, attributes, etc.

```
<?xml version="1.0" encoding="utf-8"?>
<data>
  <employee id="7" name="Georgi Panayotov">
    <birthday>2018-11-21</birthday>
    ...
  </employee>
</data>
```

- JSON (JavaScript Object Notation)

- Text format that describes data in even simpler way

```
{
  "data": [
    "employee": {
      "id": "7",
      "name": "Georgi Panayotov",
      "birthday": "2018-11-21"
    }
  ]
}
```

# Attributes in C#

- A powerful mechanism of associating metadata to code
  - As everything else in .NET attributes are classes ☺
  - Declaring custom attributes is not part the course, for more information look at [Microsoft Documentation](#)
- Usage of attributes

```
[DataContract(Name = "emp")]
public class Employee
{
    [DataMember(Name = "name")]
    public string Name { get; set; }
}
```

# Data Contract Serialization

- System.Runtime.Serialization assembly
- DataContract attribute
- DataMember attribute
- DataContractSerializer class
  - Initialization
  - Known Types
  - WriteObject
  - ReadObject
- DataContractJsonSerializer class
- Binary Serialization
- More information in [Microsoft Documentation](#)

# Newtonsoft JSON / JSON.NET

- External library added through NuGet package
  - <https://www.newtonsoft.com/json>
  - JSON.NET 11.0.2
- Serialization and deserialization of JSON
  - `string json = JsonConvert.SerializeObject(product);`
  - `Movie m = JsonConvert.DeserializeObject<Movie>(json);`
- Quite flexible for custom JSON serialization
- Popular framework used by Microsoft as well (e.g. ASP.NET Core)



Questions?

