

C# and Microsoft .NET **Strings**



Trainer: Georgi Panayotov
E-mail: smg@smg-bg.net
Phone / Viber: +359877347912



Last time...

- Bits and Bytes
- Base Data **Types**
 - Boolean
 - Integer numbers
 - Real numbers
 - Characters and Strings
 - Enumerations
 - Structs (**not really**)
 - Objects (**not really**)

- **Variables**
- Value vs. Reference Types (**not really**)
- Operators (==, +, &&, etc.)
- Console... something 😊
- Math... something 😊

....homework 😊 😊 😊

Overview

- String Data Type – Reference or Value?
- Represent a set of characters (aka text 😊)
- Strings are Unicode (UTF-16)
- Strings are **immutable**

Creation of Strings

- Initialization
- The + operator
 - +=
 - Automatic type conversion
- The String.Concat static method
- StringBuilder
 - Append, Remove, Insert, Replace methods
 - ToString() methods

Comparing Strings

- `String.IsNullOrEmpty(...)`
 - What is the default value?
 - Why can a String have null value
 - What's the difference between "" and null
- Comparing strings
 - `==`
 - `String.Equals(...)`
 - `String.Compare(...)`

Searching in Strings

- The Contains method
`"This is a text".Contains("is a")`
- The StartsWith method
`"This is a text".StartsWith("This")`
- The EndsWith method
`"This is a text".EndsWith("xt")`
- The IndexOf method
`"This is a text".IndexOf("is")`
`"This is a text".IndexOf("is", 3)`
- The LastIndexOf method
`"This is a text".LastIndexOf("is")`

Changing Strings

- The Substring() method
 - `"This is a text".Substring(0, 4)`
 - `"This is a text".Substring(4)`
- The Insert() method
 - `string line = "This is a text";`
 - `line = line.Insert(10, "short ");`
- The Remove() method
 - `string line = "This is a short text";`
 - `line = line.Remove(10, 6);`
- The Replace() method
 - `string line = "This is a short text";`
 - `line = line.Replace("short", "very long");`
- The ToLower() and ToUpper() methods
- Trimming strings
 - What is a whitespace
 - What is a trim character
 - The Trim() method
 - The TrimEnd() method
 - The TrimStart() method

Formatting

- `String.Format(...)`
- Custom String Formatting
 - C: 123.456 ("C", en-US) -> \$123.46
 - D: -1234 ("D6") -> -001234
 - F: 1234.567 ("F", en-US) -> 1234.57
 - N: 1234.567 ("N", en-US) -> 1,234.57
 - P: 1 ("P", en-US) -> 100.00 %
 - The 0, . and the # signs
- String Interpolation

Globalization

- What is Globalization
- What is a Culture
- CultureInfo
 - 10.ToString("N", CultureInfo.InvariantCulture))
- GetCultures()

Questions?

